Sixth Canadian Symposium on Instructional Technology

Halifax, Nova Scotia, Canada
May 3 – 5, 1989

Computer-Assisted Learning
Theory and Reality

Sixième symposium canadien sur la technologie pédagogique

Halifax, Nouvelle-Écosse, Canada
3 – 5 mai 1989

L'enseignement assisté par ordinateur
Théorie et réalité

PROCEEDINGS **COMPTE RENDU**

91-02986/

Sold by:

Publication Sales & Distribution
National Research Council Canada
Ottawa, ON, Canada
K1A OR6

Price: $45.00

DSS: NR15-21/1989
ISBN: 0-660-54794-5
NRCC: 30053

Vendu par:

Vente et diffusion des publications
Conseil national de recherches Canada
Ottawa, ON, Canada
K1A OR6

Prix: 45.00$

MAS: NR15-21/1989
ISBN: 0-660-54794-5
CNRC: 30053

SIXTH CANADIAN SYMPOSIUM ON INSTRUCTIONAL TECHNOLOGY
SIXIÈME SYMPOSIUM CANADIEN SUR LA TECHNOLOGIE PÉDAGOGIQUE
Halifax, NS/May 3-5, le 3-5 mai 1989

LIST OF ATTENDEES/LISTE DE PARTICIPANTS

A

ISAAC ABRAHAM
COLLEGE OF PHARMACY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

SAEED HUSAIN AL-YAMANI
DEPT OF EDUCATION
ARABIAN GULF UNIVERSITY
PO BOX 26671
MANAMA
BAHRAIN

ROBIN ALLARDYCE
31 MAJESTIC DRIVE
NEPEAN, ON
CANADA K2G 1C6

CAROLYN AMIRAULT
TOURS & HOSPITALITY DEPT
MOUNT SAINT VINCENT UNIVERSITY
166 BEDFORD HWY.
HALIFAX, NS
CANADA B3M 2J6

B

GARY BALCOMBE
WICAT CANADA INC.
1245 - 3400 DE MAISONNEUVE W.
MONTREAL, PQ
CANADA H3Z 3B8

JENNIFER BANKIER
DALHOUSIE LAW SCHOOL
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 4H9

DEBORAH BARKER
AIR CANADA BASE
MONTREAL INTERNATIONAL AIRPORT
DORVAL, PQ
CANADA H4Y 1C2

JOHN A. BARNSTEAD
DEPT OF RUSSIAN
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

ARMENOUHIE BARONIAN
CEGEP STE-FOY
2410 CHEMIN STE-FOY
STE-FOY, PQ
CANADA G1V 1T3

MUNDEN D. BATSTONE
DEPT OF CAREER DEVELOPMENT
GOV'T OF NEWFOUNDLAND
BOX 4750
ST. JOHN'S, NF
CANADA A1C 5T7

ROBERT BAUDOUIN
FAC DES SCIENCES DE L'EDUCATION
UNIVERSITE DE MONCTON
MONCTON, NB
CANADA E1A 3E9

PAUL BEAM
DEPT OF ENGLISH
UNIVERSITY OF WATERLOO
WATERLOO, ON
CANADA N2L 3G1

DIANE BEAUDRY
BELL CANADA
87 ONTARIO
BUREAU 923
MONTREAL, PQ
CANADA H2X 1Y8

HELENE BEAULIEU
735 MYRAND
APT 4
STE-FOY, PQ
CANADA G1V 2V1

JEAN-CLAUDE BELANGER
CEGEP STE-FOY
2410 CHEMIN STE-FOY
STE-FOY, PQ
CANADA G1V 1T3

WYNNE BELL
CTR FOR THE STUDY OF HUMAN LEARNING
BRUNEL UNIVERSITY
UXBRIDGE, MIDDLESEX
UK

SAMY BENGIO
DEP I.R.O.
UNIVERSITE DE MONTREAL
CP 6128, SUCC 'A'
MONTREAL, PQ
CANADA H3C 3J7

ABDALLAH BENSLIMANE
LAB D'INFORMATIQUE
COMPLEXE SCIENTIFIQUE DES CEZEAUX
UNIVERSITE DE CLERMONT II
POSTE 35-96
F-63177 AUBIERE CEDEX
FRANCE

ROBERT BIBEAU
MINISTERE DE L'EDUCATION DU QUEBEC
600 RUE FULLUM, 3E
MONTREAL, PQ
CANADA H2K 4L1

ELIZABETH R. BLACKMER
ACADIA UNIVERSITY
WOLFVILLE, NS
CANADA B0P 1X0

E.J. BLANCHARD
PLANNING & COORDINATION
SYNCRUDE CANADA LTD.
PO BOX 4023, DROP POINT 3050
FORT MCMURRAY, AB
CANADA T9H 3H5

ROBERT BLANCHARD
CONTINUING EDUCATION DIVISION
TECHNICAL UNIV OF NOVA SCOTIA
PO BOX 1000
HALIFAX, NS
CANADA B3J 2X4

ROGER BLANCHETTE
HYDRO-QUEBEC
CENTRALE NUCLEAIRE DE GENTILLY
CP 360
GENTILLY, PQ
CANADA G0X 1G0

MICHAEL R. BODEY
AT&T
PARK CENTRE PLACE
12110 PECOS STREET N
DENVER, CO
USA 80234

ANNIE BOILY
UNIVERSITE DE SHERBROOKE
2500 BOUL UNIVERSITE
SHERBROOKE, PQ
CANADA J1K 2R1

ISABELLE BORNE
E.R.E.I.
UNIVERSITE RENE DESCARTES
45 RUE DES SAINTS-PERES
F-75006 PARIS
FRANCE

ROSA MARIA BOTTINO
ISTO MATEMATICA APPLICATA
C.N.R.
VIA L.B. ALBERTI 4
16132 GENOVA
ITALY

ALONZO N. BOUCHER
MINISTERE DES COMMUNICATIONS
CP 5000
MONCTON, NB
CANADA E1C 8R3

MARIE-MICHELE BOULET
DEP D'INFORMATIQUE
UNIVERSITE LAVAL
PAVILLON POULIOT
CITE UNIVERSITAIRE
QUEBEC, PQ
CANADA G1K 7P4

KATHERINE BOWEN
DEPT OF MATH, STATISTICS & COMPUTING SCIENCE
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

TRUDY BOYLE
POLARIS LEARNING ASSOCIATES
451 KENNEDY STREET
NANAIMO, BC
CANADA V9R 2J4

JACK W. BRAHAN
DIVISION OF ELECTRICAL ENGINEERING
NATIONAL RESEARCH COUNCIL
M-50, RM 279
OTTAWA, ON
CANADA K1A 0R8

JOHN BRERETON
DEPT OF NATIONAL DEFENCE
TRAINING & DEVELOPMENT
FMO HALIFAX
HALIFAX, NS
CANADA B3K 2X0

WILLIAM R. BRODERICK
THE CERES TRUST
8 THORNDALES
BRENTWOOD, ESSEX
UK

MICHAEL BRODY
COLLEGE OF EDUCATION
UNIVERSITY OF MAINE
ORONO, ME
USA 04469

RODERICK W. BROOKES
DIRECTORATE OF INDIVIDUAL TRAINING
DEPT OF NATIONAL DEFENCE
OTTAWA, ON
CANADA K1A 0K2

DOUGLAS J. BRUBACHER
ROYAL BANK OF CANADA
315 FRONT STREET W., 13TH FL
TORONTO, ON
CANADA M5V 3A4

CATHY BRUCE-HAYTER
DEPT OF NATIONAL DEFENCE CFPS(H)
FMO HALIFAX
HALIFAX, NS
CANADA B3K 2X0

GILLES BRULE
DEP DE MATHEMATIQUE/D'INFORMATIQUE
UNIVERSITE DU QUEBEC
CP 8888, SUCC 'A'
MONTREAL, PQ
CANADA H3C 3P8

SHEILA M. BURKE
PRATT & WHITNEY CANADA
P.O. BOX 960
ENFIELD, NS
CANADA B0N 1N0

GEOFFREY V. BURSTON
CORPORATE TRAINING AND DEVELOPMENT DEPT
ONTARIO HYDRO
700 UNIVERSITY AVE., H3-A17
TORONTO, ON
CANADA M5G 1X6

CHARLES W. BURT
TRI TRAINING RESOURCES INT'L INC.
281 RETIGOUCHE ROAD
OROMOCTO, NB
CANADA E3V 2H3

DAVID L. BURT
N.S.T.C.
PO BOX 810
TRURO, NS
CANADA B2N 5G5

C

JOHN CAMERON
NORTH YORK BOARD OF EDUCATION
5050 YONGE ST
TORONTO, ON
CANADA M8W 1N4

KAREN CAMPBELL
VICTORIA GENERAL HOSPITAL
1278 TOWER ROAD
HALIFAX, NS
CANADA B3H 2Y9

GIANFRANCO CAPPELLO
IX SEMEIOTICA CHIRURGICA
UNIVERSITY OF ROME 'LA SAPIENZA'
VIA PALERMO 49
ROME I-00161
ITALY

PETER M. CAREY
AVCAT TRAINING INC.
5345 ASPEN DRIVE
W. VANCOUVER, BC
CANADA V7W 3C8

RAYMOND CARLSON
SCHOOL OF SOCIAL WORK
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

GLENN F. CARTWRIGHT
DEPT OF EDUCATIONAL PSYCHOLOGY
& COUNSELLING
MCGILL UNIVERSITY
3700 MCTAVISH STREET
MONTREAL, PQ
CANADA H3A 1Y2

CHIANG SING CHANG
DEPT OF COMPUTER SCIENCE
THE CHINESE UNIVERSITY OF
HONG KONG
SHATIN, NT
HONG KONG

MICHAEL L. CHAPMAN
ELECTROMECHANICAL DIVISION
HOLLAND COLLEGE TECH CTR
35 FIRST AVE, W ROYALTY INDUST PARK
CHARLOTTETOWN, PE
CANADA C1E 1B0

LISE CHARBONNEAU
CDN FORCES TRAINING DEVELOPMENT CTR
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

SERGE CHARBONNEAU
MIN DES AFFAIRES EXTERIEURES
125 SUSSEX, TOUR D2
OTTAWA, ON
CANADA K1A 0G2

RHONA D. CHARRON
357 CABANO STREET
LA SALLE, PQ
CANADA H8R 2M2

LARRY C. CHRISTENSEN
DEPT OF COMPUTER SCIENCE
BRIGHAM YOUNG UNIVERSITY
PROVO, UT
USA 84602

HOWARD CLARK
ARTS & ADMINISTRATION
DALHOUSIE UNIVERSITY
DALHOUSIE, NS
CANADA B3H 4H2

W.M. CODERRE
VP-MARKETING
NATIONAL RESEARCH COUNCIL
M-58
OTTAWA, ON
CANADA K1A 0R6

MICHAEL COLLINS
SCH OF GENERAL STUDIES
MEMORIAL UNIV OF NEWFOUNDLAND
ST. JOHN'S, NF
CANADA A1B 3X9

JAMES CONWAY
CDN FORCES SCH OF AEROSPACE
TECHNOLOGY & ENGINEERING
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

ELLEN CORNELL
LABOUR CANADA
165 HOTEL DE VILLE
PHASE II, 7TH FLOOR
OTTAWA, ON
CANADA K1A 0J2

ANDREW CORNWALL
N.S. DEPT OF ADVANCED EDUCATION
& JOB TRAINING
P.O. BOX 2086 STATION M
HALIFAX, NS
CANADA B3J 3B7

ASSAR CORVIN
DEPT OF PHYSIOLOGY & BIOPHYSICS
FAC OF MED, CLINICAL RES CTR
DALHOUSIE UNIVERSITY
5849 UNIVERSITY AVENUE
HALIFAX, NS
CANADA B3H 4H7

R. COULMAN
ARIES LABORATORY
DEPT OF COMPUTATIONAL SCIENCE
UNIVERSITY OF SASKATCHEWAN
SASKATOON, SK
CANADA S7N 0W0

ROBERT S.E. COWLEY
CNCP TELECOMMUNICATIONS
3300 BLOOR STREET W
STE 700
TORONTO, ON
CANADA M8X 2W9

C.G. CRAPPER
ONTARIO HYDRO
700 UNIVERSITY AVENUE
ROOM H3-B18
TORONTO, ON
CANADA M5G 1X6

MARY L. CROWLEY
OFFICE OF INSTRUCTIONAL DEVELOPMENT
& TECHNOLOGY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

D

KIM L. DALKIR
CDN WORKPLACE AUTOMATION RES CTR
DEPT OF COMMUNICATIONS
1575 BOUL CHOMEDEY
LAVAL, PQ
CANADA H7V 3X3

ERIC DAVEY
ATOMIC ENERGY OF CANADA LTD.
CHALK RIVER, ON
CANADA K0J 1J0

PAUL DAVIS
OIC TRAINING & DEVELOPMENT BRANCH
R.C.M.P.
1200 VANIER PARKWAY
OTTAWA, ON
CANADA K1A 0R2

KEITH R. DAWE
CANADIAN FORCES FLEET SCHOOL
FMO HALIFAX
HALIFAX, NS
CANADA B3K 2X0

BRIGITTE DE LA PASSARDIERE
LABORATOIRE MASI
UNIVERSITE P&M CURIE (PARIS VI)
4 PLACE JUSSIEU
F-75252 PARIS CEDEX 05
FRANCE

MICHELLE DECORTE
CEGEP STE-FOY
2410 CHEMIN STE-FOY
STE-FOY, PQ
CANADA G1V 1T3

KEVIN DELANEY
D.N.D.
COMP 58 GREEN FOREST
R.R. NO 1
LOWER SACKVILLE, NS
CANADA B4C 2S6

SHELLEY DELANO-PARKER
TRI TRAINING RESOURCES INT'L INC.
281 RESTIGOUCHE ROAD
OROMOCTO, NB
CANADA E2V 2H2

DANIELLE DERY
LAB D'INTELLIGENCE ARTIFICIELLE
EN EDUCATION
FAC DE SCIENCES DE L'EDUCATION
UNIVERSITE LAVAL
ST-RAYMOND, PQ
CANADA G1K 7P4

ALAIN DERYCKE
INSTITUT CUEEP
UNIVERSITE DES SCI/TECH DE LILLE I
BATIMENT 4
F-59655 VILLENEUVE D'ASCQ
FRANCE

LISE DESMARAIS
HUMAN RESOURCES DEVELOPMENT &
PLANNING DIRECTORATE
EMPLOYMENT & IMMIGRATION CANADA
140 PLACE DU PORTAGE, 3RD FL
HULL, PQ
CANADA K1A 0J9

CHARLES DIRKSEN
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

JOSEPH M. DONNELLY
CDN FORCES SCH OF ADMIN & LOGISTICS
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

LEE ANNE DORAN
UNIVERSITY OF NEW HAMPSHIRE
60 STRAFFORD AVE. NO J3
DURHAM, NH
USA 03824

PAUL DOUCET
DEPT OF ADVANCED EDUCATION &
TRAINING
NEW BRUNSWICK COMMUNITY COLLEGE
PO BOX 4819
DIEPPE, NB
CANADA E1A 8G1

ARTHUR J. DOYLE
COLLEGE-LEVEL & STATE SERVICES
THE COLLEGE BOARD
45 COLUMBUS AVENUE
NEW YORK, NY
USA 10023-6992

MAURICE DRAPEAU
POLVALENTE MATHIEU-MARTIN
511 RUE CHAMPLAIN
DIEPPE, NB
CANADA E1A 1P2

JULIA DRIVER
ALBERTA RESEARCH COUNCIL
6815 - 8TH STREET NE, 3RD FL
CALGARY, AB
CANADA T2E 7H7

LOUISE DUBUC
L'EVALUATION DES LOGICIELS
DIRECT DES RESSOURCES DIDACTIQUES
MINISTERE DE L'EDUCATION DU QUEBEC
1200 AVE MCGILL COLLEGE, 22E ETAGE
MONTREAL, PQ
CANADA H3B 4J8

COLETTE DUBUISSON
INFOJED
5585 MONKLAND, STE 130
MONTREAL, PQ
CANADA H4A 1E1

AUDE DUFRESNE
DEP DES SCIENCES DE LA
COMMUNICATIONS
UNIVERSITE DE MONTREAL
CP 6128, SUCC 'A'
MONTREAL, PQ
CANADA H3C 3J7

BERNARD DUMONT
TECH NOUVELLES ET ENSEIGNEMENT
INST NATIONAL DE RECH PEDAGOGIQUE
29 RUE D'ULM
F-75230 PARIS CEDEX 05
FRANCE

BRIAN DURELL
DEPT OF EDUCATIONAL PSYCHOLOGY
FACULTY OF EDUCATION
UNIVERSITY OF TORONTO
371 BLOOR STREET W
TORONTO, ON
CANADA M5S 2R7

ANNE DYCHTENBERG
CN
1060 UNIVERSITY
MONTREAL, PQ
CANADA H3B 3A2

LUCY J. DYER
DEPT OF BIOLOGY
UNIVERSITY OF NEW BRUNSWICK
BAG SERVICE #45111
FREDERICTON, NB
CANADA E3B 6E1

E

BRIAN J. EASTWOOD
DEPT OF COMMUNITY HEALTH &
EPIDEMIOLOGY
DALHOUSIE UNIVERSITY
5849 UNIVERSITY AVE
HALIFAX, NS
CANADA B3H 4H7

STEFAN EHRLICH
COMPUTER SCIENCE & MATHEMATICS
RIVIERE COLLEGE
NASHUA, NH
USA 03140

LYLE C. EMMOTT
EDUCATIONAL TECHNOLOGY CENTRE
ACCESS NETWORK
295 MIDPARK WAY SE
CALGARY, AB
CANADA T2X 2A8

F

LINDA FECZKO
ASSOCIATION OF COMPUTING
MACHINERY
11 WEST 42ND STREET, 3RD FL
NEW YORK, NY
USA 10036

PETER J. FENRICH
INTERACTIVE VIDEODISC PROJECT
BC INST OF TECHNOLOGY
4300 WILLINGDON
BURNABY, BC
CANADA V5G 3H2

PEGGY ANN FERGUSON
S.T.E.P. TRADE SCHOOL
PO BOX 497
LOWER SACKVILLE, NS
CANADA B4C 3G4

ALLEN FINLEY
DEPT OF ANAESTHESIA
IWK CHILDREN'S HOSPITAL
PO BOX 3070, STN 'S'
HALIFAX, NS
CANADA B3J 3G9

CYRIL E. FOLKES
CENTRAL NEWFOUNDLAND COMMUNITY
COLLEGE
PO BOX 413
GRAND FALLS, NF
CANADA A2A 2J8

TOM P. FORREST
DEPT OF CHEMISTRY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

CLAUDE FRASSON
DEP I.R.O.
UNIVERSITE DE MONTREAL
CP 6128, SUCC 'A'
MONTREAL, PQ
CANADA H3C 3J7

JEAN-YVES FRECHETTE
CEGEP FRANCOIS-XAVIER-GARNEAU
1334 DE REPENTIGNY
QUEBEC, PQ
CANADA G1S 1Y5

PETER L. FRENCH
DEPT OF NATIONAL DEFENCE
CANADIAN FORCES SCHOOL OF
COMMUNICATIONS & ELECTRONICS
KINGSTON, ON
CANADA K7K 5L0

LEWIS FU
REVENUE CANADA
1495 HERON ROAD
OTTAWA, ON
CANADA K1A 0L8

DOROTHY G. FUGLER JOHNSON
CONTROL DATA CANADA LTD.
110 BLOOR STREET W
TORONTO, ON
CANADA M5S 2W7

MARY F. FURLONG-MACINNIS
DEPT OF NATIONAL DEFENCE
FMO HALIFAX
HALIFAX, NS
CANADA B3K 2X0

G

PHILIP GAGNON
FACULTY OF DENTISTRY
DALHOUSIE UNIVERSITY
DENTAL BLDG., RM 3135
HALIFAX, NS
CANADA B3H 3J5

T. DEAN GAILY
CO-ORD COMPUTER BASED LEARNING
UNIVERSITY OF WESTERN ONTARIO
PHYSICS & ASTRONOMY BLDG.
LONDON, ON
CANADA N6A 3K7

HORAND J. GASSMANN
SCHOOL OF BUSINESS ADMINISTRATION
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

J. ALAN GAUDET
YOUTH INITIATIVES OFFICE
DEPT OF ADVANCED EDUCATION
& JOB TRAINING
BOX 2066, STN 'M'
HALIFAX, NS
CANADA B3J 3B7

G. GAUTHIER
DEP DE MATHEMATIQUE/D'INFORMATIQUE
UNIVERSITE DU QUEBEC
C.P. 8888 SUCC C
MONTREAL, PQ
CANADA H3C 3P8

JERRY GAUTHIER
ADMIN & SPORTS DIVISION
HOLLAND COLLEGE
140 WEYMOUTH
CHARLOTTETOWN, PE
CANADA C1A 4Z1

JAN GECSEI
UNIVERSITE DE MONTREAL
C.P. 6128 SUCC A
MONTREAL, PQ
CANADA H3C 3J7

RANDY K. GELLNER
DEPT OF NATIONAL DEFENCE
190 O'CONNOR ST
OTTAWA, ON
CANADA K2P 1T6

D. BRADLEY GERMAN
DEPT OF NATIONAL DEFENCE, CFFS(H)
C.F.B. HALIFAX
HALIFAX, NS
CANADA B3K 2X0

JULIA GIGGEY
INDUSTRIES & ECONOMIC DEVELOPMENT
DEPT OF COMMUNICATIONS
300 SLATER STREET, STE 720
OTTAWA, ON
CANADA K1A 0C8

HELENE GIROIRE
LAFORIA
UNIVERSITE PARIS VI
TOUR 45-46, 3IEME ETAGE
4 PLACE JUSSIEU
F-75252 PARIS CEDEX 05
FRANCE

SONIA GODIN
ECOLE DOLLARD DES ORMEAUX
CP 700
COURCELETTE, PQ
CANADA G0A 1R0

CHRIS GOLDING
OPEN LEARNING AGENCY
7671 ALDERBRIDGE WAY
RICHMOND, BC
CANADA V6X 1Z9

EVELINE GRAVEL
CORPORATE EDUCATION
BELL CANADA
170 LAURIER AVE., W, ROOM 204
OTTAWA, ON
CANADA K1G 3J4

ALAN GRAY
SCH FOR RESOURCE & ENVIRONMENTAL
STUDIES
DALHOUSIE UNIVERSITY
1312 ROBIE STREET
HALIFAX, NS
CANADA B3H 3E2

STEVEN H. GREVE
7306 DEWDNEY
REGINA, SK
CANADA S4T 1E6

DOUGLAS GRIFFIN
TRAINING SYSTEMS & TECHNOLOGY DIV
QUALITY ASSURANCE PROGRAM RM A1220
PUBLIC SERVICE COMMISSION
OTTAWA, ON
CANADA K1A 0M7

ALISON I. GRIFFITH
MEASUREMENT, EVALUATION & COMP APPL
ONT INST FOR STUDIES IN EDUCATION
252 BLOOR STREET W
TORONTO, ON
CANADA M5S 1V6

H

JIM W. HANCOCK
COMPUTER CENTRE
UNIVERSITY OF PRINCE EDWARD ISLAND
550 UNIVERSITY AVENUE
CHARLOTTETOWN, PE
CANADA C1A 4P3

W. DOUGLAS HAYHOE
MEASUREMENT, EVALUATION & COMPUTER APPLICATIONS
ONT INST FOR STUDIES IN EDUCATION
252 BLOOR STREET W
TORONTO, ON
CANADA M5S 1V6

JOSEPH HEATH
METAVISION INC.
307 STE-CATHERINE W.
STE 720
MONTREAL, PQ
CANADA H2X 2A4

JAMES S. HENDERSON
FACULTY OF MEDICINE
UNIVERSITY OF MANITOBA
770 BANNATYNE AVENUE
WINNIPEG, MB
CANADA R3E 0W3

WALTER HEPPLER
COUNTY OF PARKLAND PUBLIC SCHOOLS
PO BOX 250
STONY PLAIN, AB
CANADA T0E 2G0

GEORGE HILL
EMPLOYMENT & IMMIGRATION COMMISSION
PO BOX 1380
DARTMOUTH, NS
CANADA B2Y 4B9

W.D. HODGE
DEPT OF SPECIAL EDUCATION
CALIFORNIA UNIVERSITY OF PENNSYLVANIA
CALIFORNIA, PA
USA 15419

RON HOFFMAN
DEPT OF PSYCHOLOGY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

MAURICE G. HOLLINGSWORTH
REHABILITATION STUDIES
DEPT OF EDUCATIONAL PSYCHOLOGY
UNIVERSITY OF CALGARY
2500 UNIVERSITY DRIVE NW
CALGARY, AB
CANADA T2N 1N4

JEFFREY HOLMES
248 COLBY DRIVE
DARTMOUTH, NS
CANADA B2V 1K1

RAYMOND D. HOLMES
KNOWLEDGE HOUSE PUBLISHING
SUITE 201, 3845 DUTCH VILLAGE RD
HALIFAX, NS
CANADA B3L 4H9

PETER H. HOPE
DEPT OF NATIONAL DEFENCE
63 LEWIS DR
BEDFORD, NS
CANADA B4B 1C3

MONZIL HOSSAIN
KING ABDUL AZIZ NAVAL BASE
ARMED FORCES HOSPITAL-SOMC
PO BOX 413
JUBAIL 31951
SAUDI ARABIA

MICHEL HOUDE
AES
DEPT OF THE ENVIRONMENT
4905 DUFFERIN STREET
DOWNSVIEW, ON
CANADA M3H 5T4

PHILIP HOWARD
SAINT MARY'S UNIVERSITY
ROBIE ST
HALIFAX, NS
CANADA B3H 3C3

WESLEY HOWIE
NOVA SCOTIA COMMUNITY COLLEGE INST OF TECHNOLOGY
5685 LEEDS STREET
PO BOX 2210
HALIFAX, NS
CANADA B3J 3C4

MILTO HRISTOPOULOS
HOGESCHOOL VAN AMSTERDAM
SECTOR GEZONDHEIDSZORG
TER HAARSTRAAT 17
1053 LH AMSTERDAM
THE NETHERLANDS

ZORIANNA L. HYWORON
INFOTECH INC.
485 MADISON STREET
WINNIPEG, MB
CANADA R3J 1J2

I

GILLES IMBEAU
3580 BEDFORD
STE 46
MONTREAL, PQ
CANADA H3S 1G7

JACQUES IMBEAU
UNIVERSITE LAVAL
CITE UNIVERSITAIRE
1466 DE KONINCK
STE-FOY, PQ
CANADA G1K 7P4

J

CLOVIS JACOB
SCHOOL DISTRICT 13
PO BOX 637
SHEDIAC, NB
CANADA E0A 3G0

ANDRE JACQUES
UNIVERSITE DU QUEBEC
300 URSULINES
RIMOUSKI, PQ
CANADA G5L 3A1

GUY JEFFERY
TASK CONSULTING SERVICES INC
12 SUSAN PLACE
DARTMOUTH, NS
CANADA B2A 4M3

MICHAEL G. JEFFREY
COMPUTER EDUCATION
CURRICULUM DEVELOPMENT
NS DEPARTMENT OF EDUCATION
PO BOX 578
HALIFAX, NS
CANADA B3J 2S9

MICHEL JETTE
CDN FORCES SCHOOL OF AEROSPACE TECHNOLOGY & ENGINEERING
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

AMRAT JIVAN
RED RIVER COMMUNITY COLLEGE
2055 NOTRE DAME AVENUE
WINNIPEG, MB
CANADA R3H 0J9

MICHELLE JOAB
LAB D'INFO FONDAMENTALE
UNIVERSITE PARIS 6
TOURS 55-65, 305
4 PLACE JUSSIEU
F-75252 PARIS CEDEX 05
FRANCE

CARL JOHNSTON
AIR CANADA
PLACE AIR CANADA
MONTREAL, PQ
CANADA H2Z 1X5

MARLENE JONES
ADVANCED TECHNOLOGIES
ALBERTA RESEARCH COUNCIL
6815 - 8TH STREET NE
CALGARY, AB
CANADA T2E 7H7

SIMON A. JONKER
EDUCATION DEPT.
CITY OF ROTTERDAM
RODEZAND 18 (SK-427)
3011 AN ROTTERDAM
THE NETHERLANDS

GLEN JORDAN
FACULTY OF FORESTRY
UNIVERSITY OF NEW BRUNSWICK
BSN 44555
FREDERICTON, NB
CANADA E3B 6C2

CHRISTIAN JUNG
AUSSCHUSSES F. INFORMATIK DER PK-EDK
OBERFELDSTRASSE 1
CH-8500 FRAUENFELD
SWITZERLAND

K

PAUL KAISER
CANADIAN BROADCASTING CORPORATION
7925 COTE ST-LUC RD.
MONTREAL, PQ
CANADA H4W 1R5

ROBERT H. KERR
C.F.P.S.
F.M.O. HALIFAX
HALIFAX, NS
CANADA B3K 2X0

MICHAEL R. KIBBY
UNIVERSITY OF STRATHCLYDE
SCOTTISH HCI CTR., GEORGE HOUSE
36 NORTH HANOVER STREET
GLASGOW G1 2AD
UK

STAN J. KLOSEK
CUYAHOGA COMMUNITY COLLEGE
2900 COMMUNITY COLLEGE AVENUE
CLEVELAND, OH
USA 44115

FRED W. KOLBERT
CORRECTIONAL SERVICE OF CANADA
101 - 22ND STREET E
STE 405
SASKATOON, SK
CANADA S7L 0E4

RON KUBAN
ALBERTA PUBLIC SAFETY SERVICES
11612 35 AVE.
EDMONTON, AB
CANADA T6J 2X2

L

RICHARD LAFLECHE
SOCIETE RADIO-CANADA
7925 COTE ST-LUC
MONTREAL, PQ
CANADA H4W 1R5

CLAUDE LAMONTAGNE
ECOLE DE PSYCHOLOGIE
UNIVERSITE D'OTTAWA
651 CUMBERLAND
OTTAWA, ON
CANADA K1N 6N5

JEAN LANOUETTE
CENTRE FRANCO-ONTARIEN DE RESSOURCES PEDAGOGIQUES
339 RUE WILBROD
OTTAWA, ON
CANADA K1N 6M4

LOUIS E. LAPIERRE
DEP DE BIOLOGIE
UNIVERSITE DE MONCTON
MONCTON, NB
CANADA E1A 3E9

SIMON LAROSE
UNIVERSITE LAVAL
ECOLE DE PSYCHOLOGIE
PAVILLON FELIX-ANTOINE SAVARD
QUEBEC, PQ
CANADA G1K 7P4

Y.R. LAUZON
TECHNOLOGIE EDUCATIVE
FACULTE D'EDUCATION
UNIVERSITE D'OTTAWA
145 JEAN JACQUES LUSSIER
OTTAWA, ON
CANADA K1N 6N5

MARIO LAVOIE
DEP DE MATHEMATIQUE & D'INFORMATION
UNIVERSITE DU QUEBEC
300 ALLEE DES URSULINES
RIMOUSKI, PQ
CANADA G5L 3A1

COLEEN LE BLANC
INDUSTRIAL C.C. YOUTH EMPLOYMENT OPPORTUNITY
269 CHARLOTTE STREET
SYDNEY, NS
CANADA B1P 1C6

ROBERT LE BLANC
461 MARGUERITE
SITE 7, BOX 2
FOX CREEK, NB
CANADA E0A 1R0

MICHEL LEBER
CDN FORCES SCHOOL OF AEROSPACE TECHNOLOGY & ENGINEERING
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

BOB LEBLANC
UNIVERSITE DE MONCTON
MONCTON, NB
CANADA E1A 3E9

MARK D. LEBLANC
DEPT OF COMPUTER SCIENCE
UNIVERSITY OF NEW HAMPSHIRE
DURHAM, NH
USA 03824

NORMAND LEBOEUF
BELL CANADA
700 LAGAUCHETIERE O.
BUREAU 11 SUD 1
MONTREAL, PQ
CANADA H2X 1Y8

BRIAN LEES
DEPT OF COMPUTING SCIENCE
PAISLEY COLLEGE OF TECHNOLOGY
PAISLEY PA1 2BE
U.K.

JEAN-YVES LESCOP
DEP DE SCIENCES ET TECHNOLOGIE
UNIVERSITE DU QUEBEC
4835 AVE CHRISTOPHE-COLOMB
MONTREAL, PQ
CANADA H2J 4C2

JIMING LIU
MCGILL RES CTR FOR INTELLIGENT MACHINES
MCGILL UNIVERSITY
3480 UNIVERSITY STREET
MONTREAL, PQ
CANADA H3A 2A7

ROBERT A. LOCOCK
FAC OF PHARM & PHARM SCIENCES
UNIVERSITY OF ALBERTA
3118 DENTISTRY/PHARMACY CTR
EDMONTON, AB
CANADA T6G 2N8

GERALD LOGAN
DISTANCE LEARNING IN SMALL SCHOOLS PROJECT
ALBERTA EDUCATION
11160 JASPER AVE., DEVONIAN BLDG W.
EDMONTON, AB
CANADA T5K 0L2

M

JOHN S. MACDONALD
TRAINING & DEVELOPMENT
GIFFELS ASSOCIATES LTD.
30 INTERNATIONAL BLVD.
TORONTO, ON
CANADA M9W 5P3

THOMAS R. MACDONALD
EMPLOYMENT & IMMIGRATION CANADA
METROPOLITAN PLACE
99 WYSE ROAD, PO BOX 1350
DARTMOUTH, NS
CANADA B2W 2C3

MICHAEL MACNEIL
ADVANCED EDUCATION & JOB TRAINING
PO BOX 1042
SYDNEY, NS
CANADA B1P 6J7

R. NEIL MADDOX
SCHOOL OF BUSINESS
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

DONALD MANUEL
THE TRAINING GROUP
4220 - 98TH STREET
STE 202
EDMONTON, AB
CANADA T6E 6A1

EUGENE MANZIE
CDN FORCES SCHOOL OF AEROSPACE TECHNOLOGY & ENGINEERING
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

REVITAL MAROM
TVONTARIO
BOX 200 STATION Q
TORONTO, ON
CANADA M4T 2T1

TERRY E. MATTHEWS
DEPT OF ADV EDUCATION/TRAINING
NB COMMUNITY COLLEGE-MIRAMICHI
60 UNIVERSITY AVENUE
CHATHAM, NB
CANADA E1N 3W4

WENDY C. MATTHEWS
DEPT OF EDUCATION DIST 5
R.R. NO 1, BOX 214
NEWCASTLE, NB
CANADA E1V 3L8

MARGO MAYO
CDN FORCES TRAINING DEVELOMENT CTR
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

RAY MCALEESE
UNIVERSITY TEACHING CENTRE
UNIVERSITY OF ABERDEEN
ABERDEEN AB9 2UB
UK

JOHN MCCABE
DEPT OF KINESIOLOGY & PHYSICAL EDUCATION
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

GORDON MCCALLA
DEPT OF COMPUTATIONAL SCIENCE
UNIVERSITY OF SASKATCHEWAN
SASKATOON, SK
CANADA S7N 0W0

DONALD W. MCDONALD
DEPT OF NATIONAL DEFENCE
PROJECT MANAGEMENT OFFICE
CANADIAN PATROL FRIGATE
100 O'CONNOR ST
OTTAWA, ON
CANADA K1A 1G8

MARY ANN MCGARRY
RESIDENT DIRECTOR
UNIVERSITY OF MAINE
ORONO, ME
USA 04469

M. ANN MCGOUGAN
PROJECT INTERFACE
4 - 77 VICTORIA CRESCENT
NANAIMO, BC
CANADA V9R 5B9

ROBERT S. MCLEAN
MEASUREMENT, EVALUATION & COMP APPL
ONT INST FOR STUDIES IN EDUCATION
252 BLOOR STREET W
TORONTO, ON
CANADA M5S 1V6

BRYAN P. MCLENNON
INSTRUCTIONAL COMPUTING
FACULTY OF MEDICINE
DALHOUSIE UNIVERSITY
CRC BUILDING
HALIFAX, NS
CANADA B3H 4H7

ROBERT MEARNS
JOHN ABBOTT COLLEGE
BOX 2000
STE-ANNE-DE-BELLEVUE, PQ
CANADA H9X 3L9

JEFF G. MERCER
KEEWATIN TRIBAL COUNCIL
103 - 83 CHURCHILL DRIVE
THOMPSON, MB
CANADA R8N 0L6

MARC W.F. MEURRENS
COMPUTER GRAPHICS & SCIENTIFIC APPLICATIONS S.C.
51 RUE DES MOUTONS
B-1180 BRUXELLES
BELGIUM

CHET J. MICHALSKI
FACULTY OF MEDICINE
MEMORIAL UNIV OF NEWFOUNDLAND
ST. JOHN'S, NF
CANADA A1B 3V6

P. DAVID MITCHELL
CTR FOR SYSTEM RESEARCH/KNOWLEDGE ENGINEERING
GRADUATE PROGRAM - EDUC TECH
CONCORDIA UNIVERSITY
MONTREAL, PQ
CANADA H3G 1M8

CHADIA MOGHRABI
FAC DE SCIENCES ET DE GENIE
UNIVERSITE DE MONCTON
MONCTON, NB
CANADA E1A 3E9

MITCHELL MOLDOFSKY
ROYAL TRUST
95 OAKWOOD AVENUE
TORONTO, ON
CANADA M6H 2V9

RICHARD D. MONAGHAN
CLASS INC.
4128 HAMPTON
MONTREAL, PQ
CANADA H4A 2K9

JULES MORISSETTE
CTR D'ENSEIGNEMENT & DE RECH EN INFORMATIQUE CLEMENT-LOCKQUELL INC
CEGEP STE-FOY
2410 CHEMIN STE-FOY
STE-FOY, PQ
CANADA G1V 1T3

CONRAD MORNEAU
CDN FORCES SCHOOL OF AEROSPACE TECHNOLOGY & ENGINEERING
DEPT OF NATIONAL DEFENCE
CFB BORDEN
BORDEN, ON
CANADA L0M 1C0

BRUCE D. MOXLEY
FACULTY OF DENTISTRY
DALHOUSIE UNIVERSITY
DENTAL BLDG., RM 3135
HALIFAX, NS
CANADA B3H 3J5

MARIE MULDOWNEY
INST DES BANQUIERS CANADIENS
1981 AVE DU COLLEGE MCGILL
STE 675
MONTREAL, PQ
CANADA H3A 2X2

N

SOM NAIDU
DEPT OF EDUCATION
CONCORDIA UNIVERSITY
1455 DE MAISONNEUVE BLVD WEST
MONTREAL, PQ
CANADA H3G 1M8

BRUCE NAKAMURA
REVENUE CANADA TAXATION
1495 HERON ROAD, ROOM J-203
OTTAWA, ON
CANADA K1A 0L8

GEORGE NEAL
DIVISION OF ELECTRICAL ENGINEERING
NATIONAL RESEARCH COUNCIL
BLDG. M-50, ROOM 208
OTTAWA, ON
CANADA K1A 0R8

JOHN C. NESBIT
INST OF INFORMATION SCIENCES & ELECTRONICS
TSUKUBA UNIVERSITY
TSUKUBA, IBARAKI 305
JAPAN

JOHN P. NEWTON
INDUSTRIAL ACCIDENT PREVENTION ASSOCIATION
2 BLOOR STREET W., 31ST FL
TORONTO, ON
CANADA M4W 3N8

YVON NOEL
DEPT OF NATIONAL DEFENCE
CFB OTTAWA(N), DASP 4-2-2-3
BLDG 155, RM N-216
OTTAWA, ON
CANADA K1A 0K4

MARGARET NORTON
DEPT OF ADVANCED EDUCATION
56 PRINCE ARTHUR ST
AMHERST, NS
CANADA B4H 1W3

DAVID NUDELL
ST JOHN NAVAL SYSTEMS
6080 YOUNG ST
HALIFAX, NS
CANADA B3K 5L2

O

W. PHILIP O'HARA
ACADEMIC COMPUTING
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 4H8

PATRICK O'KELLY
GROUPWARE TECHNOLOGIES CORP.
1555 GLENORA DRIVE
LONDON, ON
CANADA N5X 1V6

BARBARA J. O'SHEA
SCHOOL OF OCCUPATIONAL THERAPY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

JOHN OATES
CABOT INST OF APPLIED ARTS & TECHNOLOGY
PO BOX 1693
PRINCE PHILIP DRIVE
ST. JOHN'S, NF
CANADA A1C 5P7

JULES OLIVER
INNOVATION PROGRAM
EMPLOYMENT & IMMIGRATION CANADA
PHASE IV, 4TH FL
PLACE DU PORTAGE
HULL, PQ
CANADA K1A 0J9

LESLIE H. OLIVER
ACADIA UNIVERSITY
WOLFVILLE, NS
CANADA B0P 1X0

ANNA ONUFER
CANADIAN MARCONI COMPANY
2442 TRENTON AVENUE
BOX 50
MONTREAL, PQ
CANADA H3P 1Y9

JUAN CARLOS A. OTERO
INFORMATICA/CEPET
APTDO 52042
CARACAS 1050-A
VENEZUELA

P

GARY PALEN
TRAINING TECHNOLOGY UNIT
TRANSPORT CANADA TRAINING INST
1950 MONTREAL RD
CORNWALL, ON
CANADA K6H 6L2

STEPHEN J. PALMER
ATLANTIC CANADA OPPORTUNITIES AGENCY
PO BOX 6051
MONCTON, NB
CANADA E1C 9J8

GILBERT PAQUETTE
TELE-UNIVERSITE
4835 CHRISTOPHE-COLOMB
MONTREAL, PQ
CANADA H1X 2M4

GORDON W. PATTON
SHELBURNE CAMPUS
NOVA SCOTIA COMMUNITY COLLEGE
P.O. BOX 780
SHELBOURNE, NS
CANADA B0T 1W0

VICTOR PAUL-ELIAS
EMPLOYMENT & IMMIGRATION CANADA
615 PROSPECT ST. WEST
FREDERICTON, NB
CANADA E3B 5V6

ROGER PELCHAT
LES ROBOTS PEDAGOGIQUES C.A.R.E.S., INC.
10450 MEUNIER
MONTREAL, PQ
CANADA H3L 2Z4

PIERRE PELLETIER
DEP DE TECH DE L'ENSEIGNEMENT
UNIVERSITE LAVAL
1466 DE KONINCK
QUEBEC, PQ
CANADA G1K 7P4

DOUGLAS A. PENNY
D.A. PENNY & ASSOCIATES INC.
80 GLENCAIRN AVENUE
TORONTO, ON
CANADA M4R 1M6

MARY ANN PENNY
D.A. PENNY & ASSOCIATES INC.
80 GLENCAIRN AVENUE
TORONTO, ON
CANADA M4R 1M6

MILTON W. PETRUK
DIV OF EDUCATIONAL RESEARCH SVS
FACULTY OF EDUCATION
UNIVERSITY OF ALBERTA
645 EDUCATION CENTRE S.
EDMONTON, AB
CANADA T6G 2G5

DUNCAN J. PHILLIPS
TRAINING
ST. JOHN AMBULANCE
312 LAURIER AVE., E
OTTAWA, ON
CANADA K1N 8V4

ROBERT PHILLIPS
345 HIGHLAND PARK TERRACE
AIKEN, SC
USA 29801

SUSAN E. PHILLIPS
NORTH ISLAND COLLEGE
156 MANOR DRIVE
COMOX, BC
CANADA V9N 6P7

JOCELYNE PICOT
CANADIAN WORKPLACE AUTOMATION RESEARCH CENTRE
1575 CHOMEDY BLVD.
LAVAL, PQ
CANADA H7V 2X2

MARTIN PITZ
CALGARY CATHOLIC SEPARATE SCHOOL BOARD
300 - 6TH AVENUE SE
CALGARY, AB
CANADA T2G 0G5

PAUL POSPISIL
DEPT OF COMPUTER SCIENCE
MOUNT ROYAL COLLEGE
4825 RICHARD ROAD SW
CALGARY, AB
CANADA T3E 6K6

EVA POTHIER
SEARS CANADA INC.
222 JARVIS AVENUE
TORONTO, ON
CANADA M5B 2B8

GEORGE POWER
SCHOOL OF TECHNOLOGY & VOCATIONAL TRAINING
HOLLAND COLLEGE
425 GRANVILLE STREET
SUMMERSIDE, PE
CANADA C1N 3C4

IAN E. PURKIS
DALHOUSIE UNIVERSITY
SIR CHARLES TUPPER MEDICAL BLDG
HALIFAX, NS
CANADA B3H 4H7

Q

MARTYN T. QUIGLEY
DEPT OF CURRICULUM & INSTRUCTION
MEMORIAL UNIV OF NEWFOUNDLAND
ELIZABETH AVENUE
ST. JOHN'S, NF
CANADA A1B 3X8

R

JOHN RATH-WILSON
EMPLOYMENT & IMMIGRATION CANADA
PLACE DU PORTAGE
PHASE IV, 6TH FLOOR
HULL, PQ
CANADA K1A 0J9

KEN REIMER
FIRST CLASS SYSTEMS CORPORATION
7308 - 7TH STREET NW
CALGARY, AB
CANADA T2K 1E6

T. RICH
PROGRAM SERVICES
DEPARTMENT OF EDUCATION
ROCHFORD ST., SHAW BLDG
PO BOX 2000
CHARLOTTETOWN, PE
CANADA C1A 7N8

GRIFF RICHARDS
INTERACTIVE VIDEO DISC PROJECT
B.C.I.T.
3700 WILLINGDON AVENUE
BURNABY, BC
CANADA V5G 3H2

L.R.S. RICHMOND
FACULTY OF EDUCATION
UNIVERSITY OF REGINA
REGINA, SK
CANADA S4S 0A2

TIMOTHY J. ROBERTS
SCHOOL OF BUSINESS ADMINISTRATION
DALHOUSIE UNIVERSITY
6152 COBURG RD
HALIFAX, NS
CANADA B3H 1Z5

HARV ROCKEL
ALBERTA GOVERNMENT TELEPHONES
10020 - 100 STREET
STE 9F
EDMONTON, AB
CANADA T5J 0N5

BRYAN ROLLASON
DEPT OF NATIONAL DEFENCE
CANADIAN FORCES COLLEGE
215 YONGE BLVD
TORONTO, ON
CANADA M5M 3H9

CARL D. ROSS
DEPT OF NATIONAL DEFENCE
C.F.F.S.
HALIFAX, NS
CANADA B3K 2X0

MARY L. ROSS
112-B - 18TH STREET
MANHATTAN BEACH, CA
USA 90266

JEAN-JACQUES ROUSSEAU
CONSEIL CONSULTATIF NATIONAL DES SCIENCES ET DE LA TECHNOLOGIE
240 RUE SPARKS, 8IEME O.
OTTAWA, ON
CANADA K1A 1A1

YVES ROY
WICAT CANADA INC
3400 DE MAISONNEUVE OUEST STE 1245
MONTREAL, PQ
CANADA H3Z 3B8

S

LAURAN H. SANDALS
DEPT OF EDUCATIONAL PSYCHOLOGY
UNIVERSITY OF CALGARY
2500 UNIVERSITY DRIVE NW
CALGARY, AB
CANADA T2N 1N4

RAY D. SAPP
DIT-4
NATIONAL DEFENCE HQ
101 COLONEL BY DRIVE
OTTAWA, ON
CANADA K1A 0K2

KAY GLADWELL SCHULZE
COMPUTER SCIENCE DEPT
UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MD
USA 21402

DOUG SCHWEITZER
POLARIS LEARNING ASSOCIATES
451 KENNEDY STREET
NANAIMO, BC
CANADA V9R 2J4

DONNA SHARON
PLANNING & DEVELOPMENT RESEARCH
TVONTARIO
BOX 200, STN 'Q'
TORONTO, ON
CANADA M4T 2T1

ARTHUR SHEAR
SHEAR COM. SERVICES
11 TURNBERRY LANE
HALIFAX, NS
CANADA B3M 4E5

DONALD P. SHERIDAN
COURSEWARE DEVELOPMENT PROJECT
SCH OF BUSINESS ADMINISTRATION
DALHOUSIE UNIVERSITY
6152 COBURG ROAD
HALIFAX, NS
CANADA B3H 1Z5

J.C. ERIC SHERMAN
DEPT OF NATIONAL DEFENCE
460 ELLIOT APT 102
KINGSTON, ON
CANADA K7K 3J9

RONALD D. SHINGLES
SYSTEMS ENGINEERING BRANCH (AANFDC)
TRANSPORT CANADA
11TH FLOOR, TOWER C, PLACE DE VILLE
OTTAWA, ON
CANADA K1A 0N8

JEFFREY F. SKOSNIK
BC INST OF TECHNOLOGY
3700 WILLINGDON AVENUE
BURNABY, BC
CANADA V5G 3H2

BETH J. SLEIGHTHOLM-CAIRNS
UNIVERSITY OF WESTERN ONTARIO
LONDON, ON
CANADA N6A 5C1

DONALD SMITH
MOUNT ST. VINCENT UNIVERSITY
166 BEDFORD HIGHWAY
HALIFAX, NS
CANADA B3M 2J6

LORNE R. SMITH
LEARNING SERVICES DIVISION
ONT MINISTRY OF EDUCATION
MOWAT BLOCK RM 2249
QUEEN'S PARK
TORONTO, ON
CANADA M7A 1L2

FRANCOIS ST-PIERRE
DEPT OF MEASUREMENT, EVALUATION & COMPUTER APPLICATIONS
ONT INST FOR STUDIES IN EDUCATION
252 BLOOR ST., W, RM 3-344
TORONTO, ON
CANADA M5C 2G1

ANNA STAHMER
25 GEORGE ST. SUITE 603
TORONTO, ON
CANADA M5A 4L8

DONALD E. STENTON
DEPT OF PHYSICS
UNIVERSITY OF VICTORIA
VICTORIA, BC
CANADA V8W 2Y2

PEDRO STERN
INFORMATICA/CEPET
APTDO 52042
CARACAS 1050-A
VENEZUELA

GEOFFREY W. STEVENS
OPEN LEARNING AGENCY
7071 ALDERBRIDGE WAY
RICHMOND, BC
CANADA V6X 1Z9

LESLIE STOCKHAUSEN
SCHOOL OF BUSINESS ADMINISTRATION
DALHOUSIE UNIVERSITY
6152 COBURG RD
HALIFAX, NS
CANADA B3H 1Z5

TED W. STRANG
PMO, TCCCS
DEPT OF NATIONAL DEFENCE
OTTAWA, ON
CANADA K1A 0K2

JONATHAN C. SWALLOW
DEPT OF EDUCATION
CONCORDIA UNIVERSITY
1455 DE MAISONNEUVE W., RM H-549
MONTREAL, PQ
CANADA H3G 1M8

T

RICHARD J. TALBOT
TRAINING DEPT
PARAMAX ELECTRONICS INC.
6111 ROYALMOUNT
MONTREAL, PQ
CANADA H4P 1K6

WILLIAM M. TAYLOR
CABOT INST OF APPLIED ARTS & TECHNOLOGY
PO BOX 1693
PRINCE PHILIP DRIVE
ST. JOHN'S, NF
CANADA A1C 5P1

LYNE THERIEN
DEP DE MATHEMATIQUE/D'INFORMATIQUE
UNIVERSITE DU QUEBEC
8058 ST-DENIS
MONTREAL, PQ
CANADA H2R 2G1

DAN THEUS
CANADIAN ARMED FORCES
C.F.S.C.E., VIMY POST OFFICE
CANADIAN FORCES BASE
KINGSTON, ON
CANADA K7K 5L0

IVAN TOMEK
SCHOOL OF COMPUTER SCIENCES
ACADIA UNIVERSITY
WOLFVILLE, NS
CANADA B0P 1X0

ROLF TOMLINS
CANADIAN FORCES
COMP NO 92, BARRETT SUBDIVISION
R.R. NO 1
LOWER SACKVILLE, NS
CANADA B4C 2S6

ED TOPOLSKI
DEPT OF ADVANCED EDUCATION & TRAINING
P.O. BOX 6000
FREDERICTON, NB
CANADA E3B 5H1

JOHN TRAVERS
COMPUTER INTEGRATION PROJECT
ALBERTA EDUCATION
11160 JASPER AVE., DEVONIAN BLDG
EDMONTON, AB
CANADA T5K 0L1

RICHARD TREMBLAY
DEP DES SCIENCES ECON/ADMIN
UNIVERSITE DU QUEBEC
555 BOUL DE L'UNIVERSITE
CHICOUTIMI, PQ
CANADA G7H 2B1

MICHELLE TURCOTTE
BELL CANADA
700 LaGAUCHETIERE O., 11S1
MONTREAL, PQ
CANADA H3B 4L1

U

FRANK P. URBANSKI
MANAGEMENT SYSTEMS & CONTROL
FEDERAL BUSINESS DEVELOPMENT BANK
800 VICTORIA SQUARE, 5TH FL
MONTREAL, PQ
CANADA H4Z 1L4

V

WILCO VAN DER KOOIJ
HOGESCHOOL VAN AMSTERDAM
SECTOR GEZONDHEIDSZORG
TER HAARSTRAAT 17
1053 LH AMSTERDAM
THE NETHERLANDS

JUDITH VAN DUREN
ATHABASCA UNIVERSITY
BOX 10,000
ATHABASCA, AB
CANADA T0G 2R0

BERNARD VANBRUGGHE
DEP DE BIOLOGIE
UNIVERSITE DE MONCTON
MONCTON, NB
CANADA E1A 3E9

MARCEL V.J. VEENMAN
DEPT OF PSYCHOLOGY
UNIVERSITY OF AMSTERDAM
WEESPERPLEIN 8
1018 XA AMSTERDAM
THE NETHERLANDS

DIRK M. VERMEULEN
KATIVIK SCHOOL BOARD
305 MIMOSA AVENUE
MONTREAL, PQ
CANADA H9S 3K5

PINA VETRANO
AIR CANADA BASE
MONTREAL INTERNATIONAL AIRPORT
DORVAL, PQ
CANADA H4Y 1C2

PHILIP J. VINCENT
ROCKWOOD INFORMATICS LTD
212 QUEEN ST SUITE 202
FREDERICTON, NB
CANADA E3B 1A8

ALBERT L. VOISINE
CP RAIL
PO BOX 6042, STN 'A'
MONTREAL, PQ
CANADA H3C 3E4

W

KENNETH A. WAIGHT
LAKELAND COLLEGE
VEGREVILLE CAMPUS
BOX 1468
VEGREVILLE, AB
CANADA T0B 4L0

WILLIAM S. WALSH
DEPT OF NATIONAL DEFENCE
8 CANNON CRESCENT
DARTMOUTH, NS
CANADA B3A 2H9

ALISTAIR W. WATT
NOVA SCOTIA COMMUNITY COLLEGE
INST OF TECHNOLOGY
5685 LEEDS STREET
PO BOX 2210
HALIFAX, NS
CANADA B3J 3C4

PATRICIA R. WATTERSON
JORDANHILL COLLEGE OF EDUCATION
SOUTHBRAE DRIVE
GLASGOW G13 1PP
U.K.

GRAEME WELCH
CLASS INC.
4560 STE CATHERINE ST. W APT 40
WESTMOUNT, PQ
CANADA H3Z 1S2

CHARLES J. WILLIAMS
COLLEGE OF GEOGRAPHIC SCIENCES
PO BOX 10, LAWRENCETOWN
ANNA. CO.
HALIFAX, NS
CANADA B0S 1M0

GARY WILSON
JOHN ABBOTT COLLEGE
PO BOX 2000
STE-ANNE-DE-BELLEVUE, PQ
CANADA H9X 3L9

JOHN D. WILSON
CAL STUDY COMMITTEE
NS DEPT OF ADVANCED EDUCATION & JOB TRAINING
PO BOX 2086, STN 'M'
HALIFAX, NS
CANADA B3J 3C4

DENNIS R.W. WING
NORTH ISLAND COLLEGE
36 MANOR DRIVE
COMOX, BC
CANADA V0N 6P7

GLORIA WITWICKI
COMPUTERIZED SPORT SYSTEMS GROUP
FAC OF PHYSICAL EDUCATION
UNIVERSITY OF CALGARY
2500 UNIVERSITY DRIVE NW
CALGARY, AB
CANADA T2N 1N4

HEIDI WOLTER
ARCTIC COLLEGE
KIVALLIG HALL
RANKIN INLET, NT
CANADA X0C 0G0

FAYE L. WOODMAN
FACULTY OF LAW
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 4H9

ALAN WRIGHT
OFFICE INSTR DEVELOPMENT & EDUC TECHNOLOGY
DALHOUSIE UNIVERSITY
HALIFAX, NS
CANADA B3H 3J5

KENNETH WYATT
UNIVERSITY OF NEW SOUTH WALES
PO BOX 1
KENSINGTON, NSW 2033
AUSTRALIA

Y

MAXWELL YOUNG
DEPT OF LEGAL STUDIES
COVENTRY POLYTECHNIC
PRIORY STREET
COVENTRY CV1 5FB
UK

PROCEEDINGS

of the

SIXTH CANADIAN SYMPOSIUM ON INSTRUCTIONAL TECHNOLOGY

Halifax, NS, Canada
May 3-5, 1989

Sponsored by

The Associate Committee
on Instructional Technology
of the
National Research Council Canada

. . . . . .

COMPTE RENDU

du

SIXIÈME SYMPOSIUM CANADIEN SUR LA TECHNOLOGIE PÉDAGOGIQUE

Halifax, N-É, Canada
3-5 mai 1989

Parrainé par le

Comité associé de
technologie pédagogique
du
Conseil national de recherches Canada

ACKNOWLEDGMENTS

The Organizing Committee acknowledges and is grateful for the support of the Government of Nova Scotia and the National Research Council Canada.

. . . . . .

REMERCIEMENTS

Le Comité organisateur se montre reconnaissant de l'appui du Gouvernement de la Nouvelle-Écosse et du Conseil national de recherches Canada.

ASSOCIATE COMMITTEE ON INSTRUCTIONAL TECHNOLOGY
COMITÉ ASSOCIÉ DE TECHNOLOGIE PÉDAGOGIQUE

LIST OF MEMBERS/LISTE DES MEMBRES

Dr. L. Dubuc (Chairman)
Secrétariat à la politique linguistique
1200, ave. McGill College, 22e étage
Montréal, PQ H3B 4J8
Tél: (514) 873-1105

Mr. A. Archambault
Formation
Bell Canada
87 Ontario St. W., Rm. 923
Montreal, PQ H2X 1Y8
Tel: (514) 870-7145

Dr. Robert Arn, Pres.,
ESP Incorporated
2 St. Clair Ave. West, Ste. 1701
Toronto, ON M4V 1L5
Tel: (416) 922-0087

Mr. J.W. Brahan
Div. of Electrical Engineering
National Research Council Canada
Ottawa, ON K1A OR8
Tel: (613) 993-2484

Ms E. Godfrey, Pres.
Softwords
Division of Press Porcépic
4252 Commerce Circle
Victoria, BC V8Z 4M2
Tel: (604) 727-6522

Ms Zorianna Hyworon, Pres.
InfoTech Inc.
485 Madison Street
Winnipeg, MB R3J 1J2
Tel: (204) 885-2558

Dr. Donald Manuel, Pres.
The Training Group
4220-98th Street, Ste. 202
Edmonton, AB T6E 6A1
Tel: (403) 462-6365

Dr. Gordon McCalla
Dept. of Computational Science
University of Saskatchewan
Saskatoon, SK S7N OWO
Tel: (306) 966-4902

Mr. Douglas A. Penny, Pres.
D.A. Penny and Assocs. Inc.
80 Glencairn Avenue
Toronto, ON M4R 1M8
Tel: (416) 489-4624

Dr. M. Petruk
Div. Educ. Res. Services
Faculty of Education
University of Alberta
Edmonton, AB T6G 2G5
Tel: (403) 432-3762

Dr. J. Picot
2555 Havre des Isles
Apt. 221
Chomedey
Laval, PQ H7W 4R4
Tel: (514) 686-0906

Mr. Wilson Price
Price Waterhouse
1175, avenue Lavigerie, Ste. 90
Ste-Foy, PQ G1V 4P1
Tel: (418) 658-5782

Mr. T. Rich, Chief Dir.
Programs and Services
Department of Education
Box 2000
Charlottetown, PE C1A 7N8
Tel: (902) 368-4670

Mr. P. Robinson
Training Division
ESSO Petroleum Canada
55 St. Clair Ave. W., Rm. 5129
Toronto, ON M5W 2J8
Tel: (416) 968-5260

M. J.-J. Rousseau, Asst. Sec.
National Advisory Board
on Science and Technology
235 Queen St., Rm. 836F
Ottawa, ON K1A 1A1
Tel: (613) 998-1306

Lt. Col. Ray Sapp
Dept. of National Defence
DIT-4
National Defence Headquarters
101 Colonel By Drive
Ottawa, ON K1A OK2
Tel: (613) 992-0261

Mr. Ronald D. Shingles
Systems Engineering Branch (AANFDC)
Transport Canada
110 O'Connor Street, Rm. 654
Ottawa, ON K1A ON8
Tel: (613) 993-1799

Mr. F.P. Urbanski, Sr. Vice-Pres.
Management, Systems and Control
Federal Business Development Bank
800 Victoria Square, 8th Flr.
Montreal, PQ H4Z 1L4
Tel: (514) 283-4196

Dr. D. Wing
Principal
North Island College
156 Manor Drive
Comox, BC V9N 6P7
Tel: (604) 339-8931

Mr. G. Neal (Secretary)
Industrial Liaison Office
Electrical Engineering Division
National Research Council Canada
Ottawa, ON K1A OR6
Tel: (613) 993-3478

ORGANIZING COMMITTEE/COMITÉ ORGANISATEUR

M. Petruk	University of Alberta, Edmonton, AB (Chairman/Président)
J. Brahan	National Research Council Canada/Conseil national de recherches Canada, Ottawa, ON
L. Forget	National Research Council Canada/Conseil national de recherches Canada, Ottawa, ON (Executive Secretary/Secrétaire excutif)
E. Godfrey	Softwords, A Division of Press Porcépic, Victoria, BC
D. Manuel	The Training Group, Edmonton, AB
G. Neal	National Research Council Canada/Conseil national de recherches Canada, Ottawa, ON
J. Picot	AAU Office of Educational Communications, Halifax, NS
T. Rich	PEI Department of Education, Charlottetown, PE
R. Sapp	Department of National Defence/Quartier général de la défense nationale, Ottawa, ON (Vendor Display Chairman/Président d'exposition commerciale)
D.R.W. Wing	North Island College, Comox, BC
D. Gosselin	National Research Council Canada/Conseil national de recherches Canada, Ottawa, ON (Assistant Secretary/Secrétaire adjointe)

SECRETARIAT/SÉCRETARIAT

L. Forget
National Research Council Canada/
Conseil national de recherches Canada
Ottawa, ON, Canada K1A 0R6

Telephone: (613) 993-9009
Telex: 053-3145
Facsimile: (613) 957-9828

THE ABSTRACTS WHICH APPEAR IN THIS VOLUME ARE THOSE WHICH HAVE BEEN SUBMITTED BY THE AUTHORS AND RETYPED BY THE SYMPOSIUM'S SECRETARIAT. SOME OF THE ABSTRACTS WERE SUBMITTED IN BOTH OFFICIAL LANGUAGES. THOSE THAT WERE NOT TRANSLATED BY THE AUTHOR WERE DONE BY THE TRANSLATION SECTION OF THE SECRETARY OF STATE.

. . . . . .

LES RÉSUMÉS DU PRÉSENT VOLUME ONT ÉTÉ RÉDIGÉS PAR LES AUTEURS ET REDACTYLOGRAPHIÉS PAR LE SECRÉTARIAT DU SYMPOSIUM. PLUSIEURS ONT ÉTÉ SOUMIS DANS LES DEUX LANGUES OFFICIELLES. CEUX SOUMIS DANS UNE VERSION UNILINGUE ONT ÉTÉ TRADUITS PAR LE BUREAU DES TRADUCTIONS DU SECRÉTARIAT D'ÉTAT.

TABLE OF CONTENTS/TABLE DES MATIÈRES

	Page
Acknowledgments/Remerciements	ii
List of Members/Liste des membres, ACIT/CATP	iii
Organizing Committee/Comité organisateur	v
KEYNOTE ADDRESS/EXPOSÉ PRINCIPAL	
Computer Assisted Learning and the Institution H.C. CLARK	3
PLENARY/PLENIERE	
Education and Artificial Intelligence R. McALEESE	7
Technology Transfer: The National Imperative W.M. CODERRE	14
Elements of a Provincial Educational Technology Integration Plan J. TRAVERS	20
ARTIFICIAL INTELLIGENCE/INTELLIGENCE ARTIFICIELLE	
An Inductive Learning Procedure for Adaptive Instructional Systems J.C. NESBIT, S. HUNKA	29
Developing a Learner Model for Intelligent CAI: A Systems Approach P.D. MITCHELL	34
Modélisation de l'étudiant: une approche contextuelle C. FRASSON, B. de LA PASSARDIÈRE	39
LOGIX: un système déductif et inductif à base de connaissances G. BRULÉ, L. THÉRIEN, G. LÉVESQUE	44
The CircuITS Intelligent Tutoring System R. COULMAN, S. BHUIYAN, M. MARK, B. PRASAD, M. SCHWEIGHARDT, J. GREER, G. McCALLA	49
Vers un système tutoriel intelligent enseignant une stratégie pour l'intégration mathématique H. BEAULIEU, A. BARONIAN, J.-C. BÉLANGER	54

Page

NAIADE, un système expert pour dialoguer avec l'élève
M. JOAB, E. CAUZINILLE-MARMECHE, J. MATHIEU 58

Preceptor: an Algebraic Word Problem Tutor
K. GLADWELL SCHULZE 63

Embedded Training and Intelligent User Interfaces for Application Software
R.J. CLARK 68

The Usefulness of Expert Systems as Intelligent On-the-Job Training Aids
K.L. DALKIR 70

Les systèmes diagnostiques et l'apprentissage de la langue
R.H. BROWN 75

Conception d'un module d'intervention d'un système conseiller
M.-M. BOULET, L. BARBEAU, S. SLOBODRIAN 80

The Educational Potential of Rule-based Expert Systems in the Micro Environment
J. HEATH, F. ELIE, M. HABIB 85

Structure des bases de connaissances d'un système tutoriel intelligent pouvant être utilisé pour l'apprentissage de la musique
R. TREMBLAY, B.J. FUGÈRE, L. NIEM 90

EMMA: An Intelligent Tutor
M.T. QUIGLEY 95

Automatic Error-correction in Learning LISP
R. CHARRON, F. SCHMALHOFER, G. CARTWRIGHT 99

Course Development Within a Sophisticated Computer-Managed Learning Environment
M. JONES, K. WIPOND 104

Un système tutoriel intelligent: modèle théorique et réalisation
G. GAUTHIER, G. IMBEAU 109

Adaptation dynamique de l'enseignement dans les systèmes d'EIAO: le cas du système SEISDI
A. BENSLIMANE, C.F. DUCATEAU 114

The Use of Microcomputer Based Knowledge Representation as a Tool for Learning, Teaching and Curriculum Design in Preservice Elementary Science Education Classes
M.J. BRODY, M.A. McGARRY, B. PATTERSON 119

Page

Dimensions of Intelligence in Intelligent Authoring System Functions
M. SZABO, M. ALLEN 124

A Knowledge-based Multiple-view Simulation for Training
A.M. BUELL, W.D. GODFREY, R. SHINGLES 129

Utilisation de systèmes d'EAO dans des systèmes d'EIAO
S. BENGIO, C. FRASSON, J. GECSEI 133

Innovations in Courseware Development
A. LE XUAN, R. SHINGHAL 138

Tuteurs intelligents vs systèmes pédagogiques à base de connaissances erronées: comment utiliser les erreurs produites par un apprenant
B. DUMONT 143

Spécifications d'un système informatique servant à la conception et à la réalisation de logiciels éducatifs à base de connaissances
M. LAVOIE, M. GAGNÉ, A. JACQUES 148

Thermex: système-expert appliqué à la résolution de problèmes en thermodynamique
L. THÉRIEN, B. MARCOS, A. BOILY, J. LAPOINTE, M. VEILLETTE 155

Knowledge-based or Expert Systems - A New Tool for Environmental Impact Assessment Instruction
A. GRAY, C. HICKMAN 160

Planning Proofs: A Problem Solving Coach for Geometry
S.H. GREVE 164

G: création d'un didacticiel par objets
M.W.F. MEURRENS 168

Concept Clarification and Visual Representation for a Knowledge Engineering Advisory System
D. MEILLEUR, P.D. MITCHELL 173

COMPUTER ASSISTED EDUCATION/ÉDUCATION ASSISTÉE PAR ORDINATEUR

Research and Evaluation Techniques in Computer Assisted Learning: A Quasi Meta-analysis of Contemporary Studies
L.H. SANDALS 180

Procedural and Declarative Languages: A Discussion From an Educational Outlook
R.M. BOTTINO 185

Page

Using Conceptual Information to Help Students Obtain Equations for Algebra Word Problems
M.D. LeBLANC, S. WEBER RUSSELL 190

An Interactive Environment to Teach Deductive Logic
E.H. HAEUSLER 195

Instructional Planning Based on Fuzzy Reasoning: An ICAI Component
J. LIU, G.M. BOYD 200

Computer Supported Peer Education
J. SWALLOW 205

RÉÉDUC: un programme de rééducation cognitive sur micro-ordinateur
M. PÉPIN, S. LAROSE 207

Development and Student Use of a Microcomputer-based "Smart Book" Concerning Sexually Transmitted Diseases
M.J. BRODY, D. KOPEC 212

Learning Artificial Intelligence Concepts: A Toolkit Approach
B. LEES 217

Computerized Didactic Laboratory
C. GAPPELLO, C. GABBIANELLI, M. DE PINTO, A. ARULLANI 222

Un constructeur d'énoncés de problèmes de physique
H. GIROIRE-BROUSSE 226

Model Analysis of Computer-assisted Test
J.-R. WAN, M.-J. WU 231

Establishing a Centralized University Facility for the Promotion of Instructional Technology
J.M. FRITZ, G.A. JORDAN, J. WEBSTER 234

A Model for the Application of Instructional Technology in a Literacy and Employment Preparation Program
C. LeBLANC, D. FUGLER 238

'MENU': Application Tools for an Educational Language Environment
P. BEAM, R. SIEMENS 243

CATO - Computer-assisted Topics
P.R. WATTERSON 248

The Use of Computer-assisted Learning in Distance Education
S.E. PHILLIPS 253

Page

International Cooperation in CAL
R.D. MONAGHAN, G. WELCH, L.R. WINER 258

Comment des enseignants du primaire s'approprient un logiciel éducatif
H. KAYLER 262

Homework Assignments Via the Computer
D.E. STENTON 264

The Potential for Having Large "Pop-up" Educational Menu Systems
W.D. HAYHOE 268

Quality Assurance for Computer-based Instruction
L.C. CHRISTENSEN, W. HAYS, M.R. BODEY 273

Teaching Middle-school Science with a Computer-based Integrated Learning System
T. LEVIN, R. NACHMIAS 278

EDUCATIONAL NETWORKING/CONSTITUTION D'UN RESEAU PÉDAGOGIQUE

Analyse des besoins et évaluation des réseaux locaux de microordinateurs pour la conception de centres de ressources éducatives
A. DERYCKE, C. VIEVILLE, D. POISSON 285

La communication assistée par ordinateur dans l'entreprise. Vers une stratégie de formation en cours d'emploi et de perfectionnement continu
F. HENRI, J.-Y. LESCOP 290

CO-CO in Nursing Care Planning: An Innovative Approach To Student Learning
B.J. SLEIGHTHOLM CAIRNS, P.J. O'KELLY 295

HYPERMEDIA/HYPÉRMEDIA

Animation in Science Teaching: Techniques and Examples
R. RANSOM 303

Hypercard et les fonctionnalités de l'EAO
E. de LA PASSARDIÈRE 305

Hypermédia: une technologie accessible pour le prototypage en EIAO: le système E C O
D. DÉRY, J. IMBEAU, P. DUCHASTEL 310

Page

Environnement pour l'apprentissage procédural sur HYPERCARD
A. DUFRESNE 315

Hypertext as a Medium for Interactive Learning
M.R. KIBBY, J.T. MAYES, C. KNUSSEN 320

IMPLEMENTATION ISSUES/QUESTIONS DE MISE EN OEUVRE

Review of the Use of Information Technologies in Ontario Schools
L.R. SMITH 327

The "Politics" of Introducing CAL into Law Courses and User Perspective on CAL
M. YOUNG 332

Artificial Intelligence: What Educators Need to Know
G.F. CARTWRIGHT, L. COFFIN 337

Teacher Adaptation to Computers in the Classroom
B. DURELL 341

Implementation of CAL Module for the Teaching of Chemistry
S. AL-YAMANI 345

The Courseware Development Project: Three Years On
D.P. SHERIDAN 351

Adult Literacy and Technology: An Ottawa Board of Education Storefront Project
S. NUTBROWN MASSEY 353

Incorporating Television/Video and Computers Into a Changing School Environment
D. SHARON, R. MAROM 358

An Implementation Model to Deal with Key Issues of Computers in Schools
M. SZABO, G.H. BUCK 360

Student and Educator Adaptation to Computer-assisted Learning in Canada's Arctic: The Experience in the Keewatin Region, Northwest Territories
H. WOLTER, P. FAHY, R. TAYLOR 364

Information/Automation: Computers and Contradiction in the Elementary Classroom
A.I. GRIFFITH 369

Page

INTERACTIVE VIDEO/VIDÉO INTERACTIF

Software Engineering Techniques for Interactive Videodisc Applications
D. BELL, A.K. ANWAR, P.G. BARKER 377

Acquisition de notions et changement d'attitude: une stratégie d'exploitation du vidéodisque interactif
M. DECORTE, J. MORISSETTE 382

Implementing Interactive Videodisc Instruction
G. RICHARDS 386

Designing a Computer Videodisc System to Train Forest Fire Control Personnel
L.C. EMMOTT 390

LEARNING ENVIRONMENT/MILIEUX D'APPRENTISSAGE

LOGO Twenty Years Later: From a Still Amnbiguous "Theory" to a Still Ambigtuous "Reality"
C. LAMONTAGNE' M. ROGERGE, H. BOYER 397

Un environnement objet d'exploration d'apprentissage
I. BORNE, C. GIRARDOT 402

The Computer-assisted Measurement of Musical Performance
F. ST-PIERRE 407

The Repertory Grid as a Vehicle for Exploring Teacher/Pupil Perspectives on LOGO
W. BELL, E.S. HARRI-AUGSTEIN 412

C.A.R.E.S. (Concept d'Apprentissage de la Robotique dans l'Enseignement des Sciences)
R. PELCHAT 417

Project Special Link: Telecommunications Linking Homes, Hospitals, Schools and Libraries
R. MASON, J. CAMERON 422

Preschool Children as Teachers of Computer Anxious Adults
L.A. DORAN, M.F. KALINOWSKI 427

POLICY/POLITIQUE

Strategies and Practices for International Collaboration or Education Should be Imported and Not Exported
W.R. BRODERICK, R. FOTHERGILL 433

Page

Ontario Ministry of Education Specifies its Microcomputer--
Part 3: Mid-life Crisis
R.S. McLEAN 437

Instructional Technology and the FEDS: Leadership Issues
and Initiatives
D.K. GRIFFIN 443

Assessment of Market Potential for Health Related
Courseware
Z.L. HYWORON 448

L'expérience sénégalaise d'applications pédagogiques
de l'ordinateur
H.N. SALL 453

SIMULATION

Structured and Unstructured Approaches to Simulation:
What do the Students Think?
M.V.J. VEENMAN, J.J. ELSHOUT, D.J. BIERMAN 457

FHYSIODISC - Intelligent Interactive Videosystem
W. van der KOOIJ 462

Simulation of Engineering Models by Interactive
Computer Graphics
Y.P. LEE, C.S. CHANG 465

Implications of Computer Simulations in Special Education
M. HOLLINGSWORTH 470

Integrating Technology in the Social Studies Curriculum:
The Effects of Using Multiple Applications to Enhance
Learning
G. WITWICKI, L. KATZ 475

MicroMec: un environnement intelligent à base d'objets
pour l'enseignement de la mécanique newtonienne
C. COULOMBE, A. COULOMBE, M. DÉCARY, P. DESAUTELS,
R. DESAUTELS, R. YOUNG 479

ActionMark: A Knowledge-based Program for Teaching
and Coaching Sport and Physical Activity
J.N. VICKERS, S. ABBOTT, M. MORRILL 481

Le développement d'outils de simulation pédagogique
à l'aide d'un progiciel de gestion d'entreprise
C. CAMIRAND, S. CARRIER 486

Page

TRAINING/FORMATION

Design Decisions for CBT in an Aircrew Training System
M.L. ROSS 493

Project CTRAIN: CAL Goes to Sea
R.J. TALBOT 497

The Use of CAL for Technical Training in the Canadian Navy
P.H. HOPE 501

LogiTexte: un logiciel de conception textuelle assistée par ordinateur
J.-Y. FRECHETTE, R. HAMEL 505

CAL in Architectural Education: Making the Distasteful Palatable
J. PLUME, K. WYATT 510

The Learning Centre - A New Solution to Staff Development Challenges
G.V. BURSTON, C.G. CRAPPER 515

Cours de français assisté par ordinateur (EAO) à l'intention d'un groupe de fonctionnaires
S. BEAUCHEMIN 519

The Role of Logic in Computer-assisted Instruction
J. SKOSNIK 523

Training Technology's Magic Mirror - The Computer in Computer-assisted Learning
G. PALEN 527

An Instruction/Production Model for Training Courseware
J. VAZQUEZ-ABAD, J. LeCAVALIER 529

La formation des agents d'immigration à l'heure du vidéo interactif
L. DESMARAIS 534

AUTHOR INDEX/INDEX DES AUTEURS 537

KEYNOTE ADDRESS/EXPOSÉ PRINCIPAL

COMPUTER ASSISTED LEARNING AND THE INSTITUTION

Howard C. Clark

Dalhousie University

I am most pleased to have been invited to deliver this Keynote Address and especially to welcome all of those who are visiting, to Halifax. I hope your time here is very worthwhile and enjoyable, and that the Conference is particularly interesting and successful.

The theme for the Symposium is "Computer-Assisted Learning - Theory and Reality". I am of course a university administrator although, I like to think, still very much a faculty member in terms of a continuing interest in teaching and research. But I certainly do not claim to be directly knowledgeable about Computer-Assisted Learning. So in preparation for this address, I did what any University President does and asked some of my colleagues to search out some of the recent literature for me to do some background reading. And I must say, I immediately ran into trouble - it seemed to me that I should first be clear about the precise theme of the conference and was immediately told that that might be difficult. A colleague pointed out to me that the literature refers to "computer-assisted learning", but also to "computer-assisted instruction"; to "computer-based learning" and "computer-managed learning"; to "computer-mediated learning" and machine-mediated learning"; to "computer-assisted design (of curriculum)" and "computer-designed instruction". It is not easy for a mere administrator to get a clear, precise understanding of the theme of this Symposium.

In spite of this, I can assure you that I am a believer! There are so many obvious examples, not only at Dalhousie, but at every university I know, that it is not possible to doubt the value of the computer, especially the micro-computer as a technological marvel in the improvement of instruction. It is quite clear that, of a variety of modern technologies, the computer has caught the imagination of teachers and students and that it has enormous potential for the improvement of learning and teaching. There are already many examples which demonstrate this.

Yet it seems to me that there are a number of quite major institutional issues that are not yet being addressed in most of our institutions. First, it is usual in most institutions that the initiative for projects in computer-assisted learning come from the individual or departmental level. That is, of course, as it should be - that is where creativity lies and it is at that level that teaching and learning takes place. However, this does not make for high efficiency. Institutions, whether they be colleges or universities, are these days large and fragmented organizations. Knowledge of one individual or departmental initiative does not usually percolate through the entire institution with any great efficiency. And even within one discipline, there is often not a very effective flow of information from one institution to others of new work in computer-assisted learning. There is still a great tendency to re-invent the wheel.

Second, because computer-assisted learning is still a "local" initiative within the institution, we are not doing, nor are we yet capable of doing, much of a job in relation to planning. Many of you I am sure would agree that computer-assisted learning is likely to become more and more important. Yet we are still grappling to find the best ways institutionally of exposing students to a wide range of relevant computer applications. We cannot yet agree easily on what computer literacy means, as presumably it must apply to all of our students, and we have great difficulty in determining what are the desirable computing skills that students should acquire before graduation. This means that even in institutions where there is

a "computing czar", his or her role is often an empirical and arbitrary one. The task is to ensure that faculty members and departments have access to micro-computer pools and other computing resources even though there are no recognized or agreed objectives or purposes for which these facilities will be used. There are certainly no agreed educational objectives. Indeed, at a time when post-secondary education in Canada is so demonstrably under-resourced, it is quite remarkable how we allocate resources for computing. It would be unthinkable, in any other area of activity, to so arbitrarily allocate resources as we do to computers, often for computer-assisted learning. To be committing resources to computers for computer-assisted learning without any sense of objectives is really quite remarkable!

Third, planning is preparation for the future. But again because so many of the initiatives for computer-assisted learning are localized within the institution, little forward-looking planning is done. It is easy, of course, to say that, yet since we do not know the capabilities of the next several generations of computers, planning is incredibly difficult. But even if we have difficulty in doing this, we should at least be making some better provisions than we do at present in terms of planning resource allocations for the future. I am not sure that any of us know how to do this, yet we do live from hand to mouth in the provision of resources for computer-assisted learning, and we somehow need to do better in future.

Fourth, evaluation of education programmes is not a popular concept, at least in post-secondary education in Canada. In relation to computer-assisted learning we have a very large task in front of us. Many computer-assisted learning projects are still introduced without incorporating any concept of self-evaluation. This is not surprising since we are still not particularly good at the definition of educational objectives and hence often have little against which to measure success or failure. I suspect that far too many computer-assisted learning projects have been initiated, have operated with great enthusiasm for some time and have been ended as enthusiasm has waned, but with little or no effort to gain educational understanding through evaluation from them. It seems to me that in addition to planning there should be an institutional requirement for evaluation built into all computer-assisted learning projects.

I am sure you will appreciate that these are the thoughts of one preoccupied with administrative matters. Yet the need for us all in higher education to engage in a more substantial debate on the definition of educational objectives, to plan and to allocate resources wisely and efficiently, and to evaluate critically all of our activities including computer-assisted learning, will increase not diminish in future.

PLENARY/PLÉNIÈRE

EDUCATION AND ARTIFICIAL INTELLIGENCE

R. McAleese
University Teaching Centre
University of Aberdeen
Aberdeen, UK

This paper surveys the contribution of artificial intelligence (AI) to instructional design. It concludes with a set of questions that designers should pose of AI research. The paper examines the concept of AI critically and gives examples of natural intelligence and artificial intelligence.

Instructional designers reflect the style and type of instruction required by a wide variety of clients. School systems require more objective-driven learning; work-place training requires more cost-effective CBT/(CAI) programs. Education, in the widest sense, requires instruction to meet the immediate needs of learners, taking into account their existing knowledge, skills and attitudes. As designers we have a general model of the good tutor. Such a tutor is, among other things, adaptive and knowledgeable. The ability to be adaptable and knowledgeable are two of the requirements that designers need of CBT. It is in these areas that artificial intelligence helps. Computer based training that more accurately reflects the way learners learn and the way the learner expects the "tutor" to behave has been aided by AI research into misunderstandings, problem solving, expertise and user models. The greatest contribution that AI has made has been to put the "content" back into CBT by emphasizing the encoding of domain knowledge.

INSTRUCTION ET INTELLIGENCE ARTIFICIELLE

Ce document présente un apperçu de la contribution de l'intelligence artificielle (IA) à la stratégie de développement et énonce un ensemble de questions que les concepteurs devraient se poser concernant l'intelligence artificielle. Ce document contient une analyse critique de la notion d'IA et énonce des exemples d'intelligence naturelle et d'intelligence artificielle.

Les stratégies de développement élaborées par les concepteurs sont basées sur le style et le type d'instruction voulus par des clients très variés. Les systèmes scolaires prônent un apprentissage davantage orienté objets alors que le milieu de travail désire des programmes EAO (AAO) plus rentables. L'instruction, prise dans son sens le plus large, doit répondre aux besoins immédiats des élèves et tenir compte de leurs compétences, aptitudes et attitudes actuelles. Nous, les concepteurs, avons une idée générale de ce que doit être un bon tuteur: il doit être, entre autres qualités, adaptable et bien informé. Ces qualités sont deux des critères voulus de l'EAO par les concepteurs. L'intelligence artificielle présente des avantages dans ces deux domaines. Les travaux de recherche en intelligence artificielle, dans les domaines des malentendus, de la résolution de problèmes, de l'expertise et de modélisation par l'utilisateur, ont permis de mettre au point un enseignement assisté par ordinateur qui correspond avec plus de précision aux modalités d'apprentissage et aux attentes des élèves quant au comportement du "tuteur". La plus grande contribution de l'intelligence artificielle a été de ramener le "contenu" dans l'EAO en accordant une grande importance au codage des connaissances des domaines.

EDUCATION AND ARTIFICIAL INTELLIGENCE

R. McAleese

University Teaching Centre
University of Aberdeen
Aberdeen AB9 2UB
Scotland

INTRODUCTION

It is difficult to judge the effect that *the* major development in Computing Science over the past twenty years has made to Education. No one can make such a judgement without feeling that any judgement is always premature and that any predictions will inevitably be found wanting. There are many articulate critics to the excesses of claims made for AI (eg Winograd and Flores, 1986). Yet is is important for us to consider again the contribution of artificial intelligence (AI) to education. At the 5th Conference held in Ottawa, seventeen papers were grouped under the heading of artificial intelligence. John Self, in the Keynote address, suggested that "AI products are the latest technology wheeled in to provide short-term solutions to long-term problems" (Self, 1986) He rightly concluded that "using AI products is not AI". In practice this emphasis has not changed. This new look at AI will not differ too much from that conclusion. Authors at that time did review in the broadest sense what AI had contributed; yet many of the presentations were myopic with regard to the full influence of AI. On reflection only Mitchell and Dalkir (1986) were able to see beyond the immediate possibilities of languages (eg Prolog) and dialogues to the influence of artificial INTELLIGENCE on education. However, their ideas, based on the seminal work of Gordon Pask (Pask, 1976) is not yet and may never be fully implemented.

Earlier, O'Shea and Self (1983) in their key book conclude that research on computer based tutoring is important because it *leads to a deeper understanding* of tutoring by humans. They say "..it is often the case that the attempt to automate an activity forces a better understanding of the activity it's self..". Attempting to evaluate any process does require us to build working examples. This paper will be examining several such working examples.

A starting point for the discussion can be to RE-define what can be meant by *artificial* and *intelligence*.

ARTIFICIAL INTELLIGENCE

Wild claims for the emergence of the super-intelligent machine are misplaced. Pundits such as Clive Sinclair (in the UK) recently suggested that by 2020 machines would surpass the intelligence of humans and, further, go on expanding their capabilities with no obvious limits!! Even Turing was prone to claim that in due course machines would "think". Such predictions have little to do with the careful study of the interaction between man and machine. Artificial intelligence, in education, is about ways of describing educational processes - thinking, understanding, reasoning learning and recognition (O'Shea and Self, 1983). AI is NOT about making wild claims. Education, being an eclectic subject, draws heavily on the ideas of subject domains to further its methods, content and aims. Although intelligence and its measurement is usually claimed by the psychologists, it is relevant to education when described in terms of learning and adaption.

Artificial vs Natural Intelligence

In order to evaluate the contribution of AI to education let us examine some basic assumptions. Let us begin, if we can, by setting down *benchmarks* that will enable us to judge the effectiveness of any claims made for AI. AI must have to do with "intelligence" To attempt to create artificially something called "intelligence" would suggest that, whatever it is, it is important. Why else would be wish to REcreate it?

What is the nature of intelligence? To make any progress we must clarify any distinctions that exist between forms of behaviour or adaption that characterise intelligence from those that characterise "non-intelligence". A tree is not intelligent, whereas university professors are intelligent. A thermostat is not intelligent whereas an engineer who can control a nuclear plant is intelligent. By extension, machines that select responses at random from a set of possible answers or have no extant model of the task or user would not be considered "intelligent". How about a teacher that uses prescribed curricular materials and simply tells his/ her class to follow work card 25a? Any artefact or process that *seems to behave* in an intelligent way can *not* be credited with intelligence. Behaviour alone proves very little.

Passing the Turing test may be a necessary but not a sufficient condition. The concept of intelligence in relation to humans is so obvious, or so it seems, that it is often difficult to disentangle the pejorative nature of an"intelligent" human from an attribute of all humans. A common assumption is that any human activity is "intelligent". Nothing can be further from the case. Intelligence must not be given solely to humans in so far as we can equate intelligence with adaptability. An adaptable machine, in that sense alone, is intelligent.

Another bench mark might be expertise. To understand the impact of AI on education it is necessary to look at expertise in conjunction with intelligence. If intelligence can be thought of as adaptability; expertise may be thought of as knowledgeability .

The nature of expertise

Expertise is the ability to perform a variety of cognitive or motor tasks to a standard that is beyond the reach of the normal performer. A surgeon who can undertake a complex operation is an expert with regard to the operation. He or she may not be an expert with regard to car maintenance. A teacher who is skilled at helping children with learning difficulties overcome such handicaps is an expert. He or she is not an expert with regard to surgery or car maintenance. In other words expertise is domain specific. There can be no real concept of "general" expertise. Therefore an AI system that attempts to mimic an expert only does so with regard to a small and carefully defined set of domain specific tasks.

Natural Intelligence

When we say that an animal is "intelligent" we tend to be thinking of the way animals behave. Behaviour such as hygiene, reproduction, feeding, etc. Such activities might well be seen as "intelligent" in that they enable the animal to survive in a particular environment. Yet they are no more intelligent than pure physiological behaviour such as digestion, breathing or cell division.

Artificial Intelligence: definitions

AI refers to the study of that set of procedures that attempt to mimic the adaptability of a human tutor. AI is an attempt to simulate the natural tutor. It is not super intelligence; it is not knowledge *per se*. AI is bounded by the ways we see intelligence and human expertise. AI is a set of procedures that help the designer produce effective and efficient tutorial programs.

Some examples of AI applications in education

It is difficult to select key developments in artificial intelligence. There have been a number of projects that have contributed to our knowledge base. However, if we examine these in isolation we can see some of the important features.

Some twenty five years ago the National Science Foundation of America funded two projects in computer assisted learning to support what some have felt was then a faltering attempt to make computers behave like tutors. Time-shared Interactive Computer Controlled Information Television (TICCIT) was undertaken by the Mitre Corporation. Programmed Logic for Automated Teaching Operation (PLATO) was based in the Computer-based Education Research Laboratory in the University of Illinois. No one now would claim that TICCIT or PLATO were at the time artificial intelligence projects. Yet they must be seen as mile stones in the development of technology and interfaces to be subsequently used by other developments.

TICCIT was aimed at achieving cost efficient tutoring. It was concerned with replacing traditional instruction. The "industrial" model of course construction was used by Mitre. Course teams consisting of designers and technicians aimed to produce cost-efficient concept learning and testing familiar to trainers. The model adopted was explicit, but in hind-sight was not as well substantiated by research on cognitive process as it might have been. Work led to a model of the tutor as an advisor, diagnostician and problem solver. Mitre's own evaluation (Mitre Corporation, 1974) was centred on the "potential" of the developments and did not really address the appropriateness of the cognitive model. The instructional model used, was *embedded* within the system.

Among a number of aims PLATO was concerned with networking of technical facilities and the provision of a simple authoring language (TUTOR). The technology required to support the complicated architecture was a major development. Yet the authoring language TUTOR has been the lasting influence of PLATO. As an interface that allowed teachers and tutors to construct their own programs it permitted a wide variety of control mechanisms to be built into tutoring. The dominant instructional model was still that of the teacher/ designer not the system architect.

SCHOLAR was a project that aimed to build a program that reviewed learner's knowledge of the geography of North America. Carbonell (1970) claims that it the first attempt to teach in the"same way as human teachers do". SCHOLAR was a *generative* system in terms of CBL (O'Shea and Self, 1983) in that it builds responses and questions rather than having them hard coded. In essence SCHOLAR had no model of the user or his/her understanding. It had a limited model of the subject domain. The underlying architecture was of a semantic net of nodes and links. As such 'the knowledge representation problem' (as it has come to be known) entered the arena. In SCHOLAR the networks were general purpose networks. Once the hierarchical structure of knowledge had been established, different subjects could be developed.

WHY (Collins, 1977) depends on Socratic tutoring to achieve its goals. In this the tutor does not know domain expertise or 'teach' a subject by exposition or explanation but by successive questioning in order to generate principles based on specific cases. As such it can capture a tutorial strategy based on Socratic rules or inquiry teaching. There is a curious and unresolved attraction to Socratic coaching in AI. It is as if such coaching can be successful in the absence of an effective domain model. However true Socratic coaching is not the same as content-free, non-directive dialogues that can be produced by simple versions of programs such as ELIZA.

SOPHIE (Brown, Burton and Bell, 1975) explored the nature of dialogues between learners and the computer system. In essence the computer had to have a sophisticated linguistic capability. The intention or meaning of an input (sentence or utterance) from a learner had be to inferred by the system. SOPHIE permitted a compromise between requiring many input sentences which would clarify the intention of the learner but would make the dialogue more complex and partially ambiguous inputs whose intention (or meaning) had to be inferred. One such problem is that of *ellipsis*; the inference of "missing" concepts from the input of the learner. Such missing intentions are a telling test for any new AI programs. Hymans use a wide variety of ambiguous linguistic techniques; metaphors, analogies, etc. Any intelligent tutoring system must be able to infer intent from a metaphor or from techniques such as elipsis. AI techniques were developed to infer missing concepts and to resolve ambiguities. This gave a technical problem for the developers as the processing time for inference to occur had to be kept to a minimum to maintain the flow of the dialogue. SOPHIE showed that it was possible to simulate a dialogue that learners would find "natural". It did not of course address the question on an intelligent system; rather it provided a system that had an interface that simulated human conversation. More recent developments with SOPHIE (SOPHIE-III: Brown *et al* , 1982) has extended its ability to intervene in the learning process. Intervention in the learning process is a key element in effective tutoring. In essence, SOPHIE has extended our AI knowledge into mental models and qualitative reasoning with an emphasis on domain specific knowledge representation.

A group of significant research papers and project reports surround the BUGGY stable (eg Burton, 1982). The project aimed at including in a tutoring system, an explicit model of incorrect knowledge. If it is possible to catalogue the errors of a task or set of tasks and provide such data as a database of 'bugs'. An understanding of possible errors and their correction has an important outcome for remedial action. What to present next, the construction of a dialogue or the topic to be covered can be determined by the nature of learner "bugs". BUGGY has contributed substantial output on problem-solving and learning models. PIXIE, which grew out of LMS, attempts a generative theory of bugs. It attempts to generate 'mal-rules' by the backward reconstruction of problem-solving errors. In so doing it builds a model of the user. Modelling of the user in terms of task and cognitive process is a dominant feature of AI research. As such it is relevant to expertise but "knowledgeability" is different in the first instance.

If there is any doubt about the preceding developments, MYCIN was an artificial intelligence program (or genus of programs) (Shortliffe, 1976). It was developed in the expert system strand of artificial intelligence developments. The problem was to produce an expert system that would be able to diagnose bacterial infections in the blood and suggest appropriate treatment. It was designed to be used by clinicians. The knowledge base was constructed in a series of production rules that

represented knowledge in a series if IF-THEN-ELSE statements. Such production systems can run forward through a series of steps or backtrack in order to infer a possible course of action. In fact MYCIN used a backtracking procedure in order to focus the solution, rather than to seem to be jumping from question to question. MYCIN had an ability to explain its diagnosis. To do this it was able to say which of the rules from which a conclusion had been drawn. MYCIN represented knowledge as a series of production rules; it ran a production system of rules backward to enable it to carry out plausible reasoning for the clinician and when required give explanations. MYCIN was able to simulate the expert and therefore a learner by observation could learn by imitation. To some extent MYCIN could behave like a tutor in giving explanations of the different steps in a complex process and allowing the learner to attempt solutions and then commenting on these solutions in relation to "expert" solutions. GUIDON was a development in which a teaching program was developed to transfer the expertise of MYCIN to the novice doctor. In order to carry out effective dialogues with learners, the program has to know something about the nature of dialogues as well as "knowing" about biological infections. GUIDON kept a separation between tutorial knowledge and medical knowledge. Further technical developments developed graphical interfaces to the knowledge bases and helped tutorial interaction (GUIDON-WATCH)

The systems described here can usefully be seen to fall into two categories. Those that address the question of how to model expertise (eg SOPHIE, GUIDON) and the nature of the appropriate intervention or response to the learner (eg WEST). The distinction between understanding and intervention is crucial in effective tutoring and instructional design.

In summary, one can see that large scale tutor-simulation programs and projects in traditional CBT such as TICCIT and PLATO provided a basis on which the AI community were able to gain links to the education community. Further LOGO, although I have not discussed it here, has provided a widely discussed and much maligned learning environment (Papert, 1980). AI has provided many notable players and concepts. SCHOLAR and WHY have enhanced what we know about tutorial-dialogues. SHRDLU with SOPHIE gives us benchmarks for natural language communication with an understanding of machines. SOPHIE simulated human-like reasoning and given us insight into reasoning; while BUGGY and PIXIE have made explicit the nature of "cognitive-bugs" and "mal-rules" GUIDON, MYCIN and NEOMYCIN gave a way of representing expertise and rule-based reasoning. All of these players have enriched the designers landscape.

The Importance of MODELS

Central to the success of artificial intelligence has been the building of models of users and systems. The engine that drives an adaptive system is akin to a prescriptive model of the essential features of that system. Models of behaviour allow systems to be adaptive. Without a model of pupils and their needs a teacher is unable to vary the level or type of instructions. Success with models has centred on explicating the nature of models (Johnson-Laird, 1983) and building operational systems that incorporate these features (Murray, 1988). A model is built on information about the way individuals perform in different circumstances. TUTOR and other authoring languages (eg PILOT, MICROTEXT) gave some help to the designer in logging user interaction. Such datalogs or user profiles are insufficient to do more than describe what has happened in a tutorial. What is required is a set of procedures that operate during a tutorial to make the dialogue, and the topics studied adaptable to the needs and abilities of the user. In one case, GUIDON uses an overlay model of the user in terms of the experts knowledge.

What are education's needs for AI?

Education has come to make use of AI for a variety of reasons. First it is there. Education and more particularly educators are just like any other group with society. They are fascinated by the ability of man to use machines. The relation between man and machine may be a love-hate symbiosis. It has pervaded teaching, tutoring and learning since medieval times. Second, education requires to adapt to the changing needs of its clients.

Different needs of training and education

It may be that there are differences between training and education that determine how AI has adapted to different market places. School-systems require more objective-driven learning; work-place training requires more cost-effective CBT-programs. Education, in the widest sense, requires instruction to meet the immediate needs of learners and take into account their existing knowledge, skills and attitudes. Yet education is more about general training as opposed to competencies in agreed skills or performance at set levels. This is how one can distinguish training from education. As a debate the training vs education schism is only interesting and useful when it highlights the goals and process by which such developments as AI can be examined.

WHAT IS THE GOOD TUTOR?

Tutors are patient; tutors work with learners; tutors help learners to understand. Tutors behave differently with different learners in different circumstances with different topics. Tutors know when to intervene and when to allow the learner to discover for themselves. The nature of tutoring, according to these assertions, is a *human* activity. Such assertions are chosen to highlight the human nature of tutoring. However, we can learn from the world around us and therefore we can learn from information that is presented by a machine. Learning from experiences is not the same as being taught.

What can AI contribute to Education?

AI has given us a set of tools that can be effectively used to design better instruction. (see below) Three should be noted.

a. User Models
b. Natural Language Interfaces.
c. Knowledge Encoding.

As a tool kit these developments can be taken in isolation or in concert. It is possible to develop more adaptive programs by building effective embedded user models. Such programs may have direct manipulation interfaces of objects as opposed to a command line interface based on natural language. Not all developments need be involved in the encoding of knowledge. Adaptive tutors may revert to the content free dialogue of ELIZA and the philosophy of non-directive tutorials. Of the tools that have been developed models and interfaces have been referred to above. Knowledge encoding merits further comment.

The encoding of domain knowledge

Without knowledge it is impossible to think of a fully intelligent tutor. Tutors KNOW things! Not only do they know how to tutor and the nature of their subjects they know about subject domains. This claim is becoming progressively more difficult to sustain with the explosion of scientific knowledge and the production of generic training packages. Nevertheless it is important to note the importance of knowledge elicitation and representation techniques. Gaines and Boose(1988) , in a recent survey of the state of the art in knowledge elicitation, indicate the wide range of problems in helping experts give up their secrets.

A New set of procedures for CBT

Among the most important of the computing products that AI has given to CBT has been developments in object-oriented programming.

Smalltalk (Goldberg, 1977) has been the best documented and implemented set of techniques for unifying the communication of programming intent into computer code. Smalltalk emerged from Xerox PARC and the Dynabook research programme. It draws on the development of LOGO and can be seen in developments such as the hypertext system called NoteCards (McAleese, 1987) and learning simulations such as Thinglab (Borning, 1981). Object-oriented techniques and high quality graphic interfaces has made other AI developments such as GUIDON more user friendly.

EDUCATION

So far, I have concentrated on the nature of AI. Education, as a part of our society has changed dramatically over the last fifty years. Not all of the changes have been vectored on one direction. Many changes are cyclical and this has to be appreciated when we come to drawing on AI for inspiration and help. The emergence of the individual and the emphasis of individual differences has been achieved despite the power of the CLASS and the CLASSroom in education. In part this has been made possible due to technology (ie computers) but also due to the emphasis of humanist and individualist approaches to learning. Changes are not always as suggested in one direction. The emphasis on the individual and individual based goals and individual needs that brought about the Kellar Plan, PSI and general theories of individualisation (Postlethwait, 1988) has given way to the needs of society for specified skills and abilities. In these changes instructional design has been able to develop a robust set of procedures that help the designer and learner.

What is instructional design?

There are many instructional design models ranging from the precise and prescriptive analysis of the training approach by Romiszowski (1983) to the more psychological approach of Gagné (1965). Instructional design is a determining factor in analysing the contribution of AI to education. Instructional design allows the teacher, the designer and the author (see below) to communicate with the learner through the interface of the CBT system. As such it is a set of procedures that includes programming techniques and languages (eg Prolog, POP-11), curricular assumptions (eg the spiral curriculum), and authoring languages (eg TUTOR, PILOT) to implement the general educational assumption of the teacher or trainer. Different design models will determine the degree to which the learner is in control or the teacher is in control over what is learned and how it is learned. Design determines the contextual background within which AI can operate.

The nature of technology and education

Technology of education is not the technology of artefacts. Rather technology in the true sense of a system of interdependent parts. Technology in education is concerned with artefacts and the way learners and designers use such devices. As such it is important for human factors and the emerging discipline of human computer interfacing to make the learning/ designing interface as effective and efficient as possible. There are many problems to be resolved. For instance it is not yet clear whether computer systems can model the wide variety of human styles of interaction as efficiently as training humans to learn to use new machines. The needs of learners, designers and teachers has to be taken into account.

The needs of learners

Learners need to be able to meet their objectives and those of the system within which they operate with a minimum of errors. Interacting with computers and other technical systems (eg teletext) requires learning many new and different strategies.

The needs of designers/ teachers

The relationship of teachers, designers, authors and learners is a complex one. Figure 1 indicates the way the participants in this system interact with each other. In any tutoring system there are four participants; the Teacher (with domain knowledge), the Designer (with design knowledge) the Author and the Learner. (NB: in some systems the participants are in fact "roles" played by the same individual, ie teacher as designer, designer as domain expert etc) As a result of interactions between the teacher, the designer and the author an intelligent AI tutoring system is built. The learner in interacting with it, is interacting with a series of models of the task, the domain, etc. Information or feedback is available to each of the participants. As such the system is held in a stable state by the updating of the AI system by any one of the participants.

THE FUTURE

If AI is to make further contributions to education it must be able to take on board some of the changes in our educational system and the technical developments in computers. I have alluded to the changes in education that are taking place. The technology driven developments are easy to foresee and impossible to be certain about. Of course there will be more,

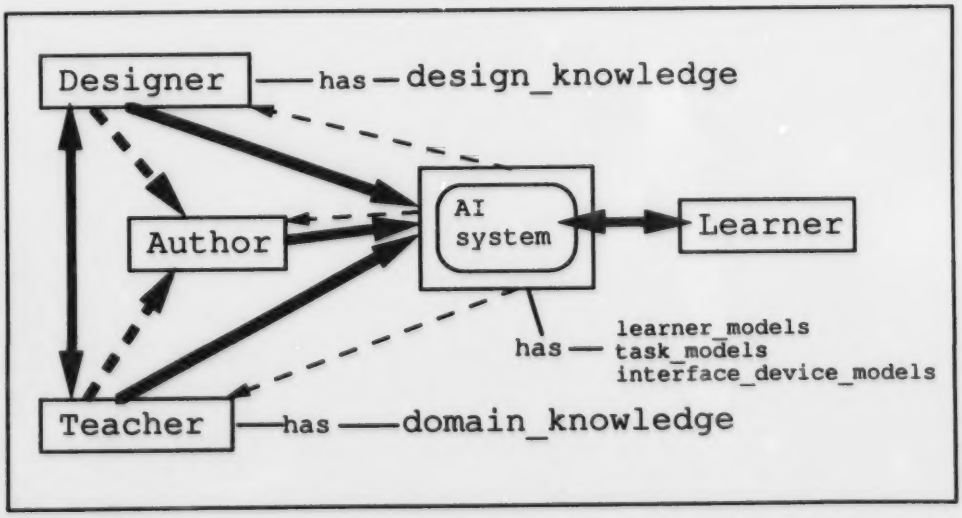


Figure 1: Interactions Between Teacher, Designer, Author, Learner and an AI System

faster and better computers. Human factors engineering will make computer interfacing more "friendly"; ie easier to operate and easier for the user to build system models. In practical terms AI will have to live with developments in new areas such as hypermedia and better attempts to model expertise.

AI and Hypermedia

Hypermedia has taken the design of instruction from the linear straight jacket of programmed learning into the multi-dimensional world of semantic nets (McAleese, (1989). Non-sequential access to information and the use of mass storage devices such as the CD-ROM and WORMs is giving a new set of system models for the designer. The locus of understanding and modelling shifts when there is no inherent sequence or structure on instruction. Hypermedia systems emphasise the needs of the user in determining where to go and what to learn. As such hypermedia and AI seem to collide with each other. While AI has been attempting to model users and develop adaptive tutorials, hypermedia and projects such as Xanadu (Nelson (198?) have used the power of the networked computer to provide individualisation through variety as opposed to individualisation through precise modelling. Hypermedia does not run counter to AI but the development of AI techniques and the growing power of computing systems based on parallel processing will emphasise the power of technical developments against the precision of modelling complex processes.

Questions from Education for AI

In conclusion I will look briefly at the nature of questions that can be posed of AI if we are to develop an ongoing relationship. It would seem at times that AI has begun to meet the expectations of Samuel Butler (1927) when he wrote " ...man's very soul is due to machines..." Yet we are still a long way from achieving the symbiotic relationship implied here. If one looks at AI from the point of view of a classroom teacher then only two questions arise:

a. How can I more ***effectively*** design instruction?
b. How can I more ***efficiently*** design instruction?

Such questions are not those answered in the past ten years by the AI community. As such these questions have been asked for twenty-five years and were, as suggested, more relevant to PLATO than SOPHIE-III.

More relevant questions relate to the cognitive processes in tutoring, learning and skill acquisition. For example:

c. What tutorial model is embedded in the system?
d. Can domain expertise be encoded?
e. Can mal-rules be encoded?
f. To what extent does the system build a model of the tutorial dialogue?

When instructional designers look to AI research for inspiration they must look with AI-tinted spectacles. AI research does not answer the macro question of effectiveness or efficiency. If it were ever possible to answer such questions the answers are forgotten and the time is long past when it were possible to suggest that one technology based form of instruction was better than or less costly than some other form of tutoring.

AI research has made education question some of its assumptions about the nature of the interaction between teacher and taught. In this area alone it has given an unparalleled stimulus to designers to look at dialogues and interfaces. It has enabled us to look more objectively at what we do as teachers and indicate some evaluation criteria. Tools have been made available to us to help with the job of preparing instruction. The locus of control in tutoring has not shifted to the computer or the computer system, rather teachers and designers have benefited from insights into their jobs by looking at the way AI has attempted to build working models of human-tutoring systems.

It is impossible to see how things will develop without an eye to the commercial exploitation of information technology and computer based systems that was given such emphasis by the Japanese Fifth Generation Project. However these developments and the stunning new ideas about massively parallel computers will have little impact in the classrooms of to-day, or the immediate to-morrow. We will have to wait for the needs of the learner and the teacher to outweigh technological costs. AI is giving weight to that cause by providing adaptable computer-based tutors

Finally, AI has not been without its practical or philosophical critics; in particular the devastating phemenological critique by Dreyfus (1972). The balanced view of such work (which I have chosen to ignore here as we should take a more pragmatic view in instructional technology) is that we are still coming to terms with the nature of intentionality and the separation of the mind from the brain. Such debates have more to do with the staff room than the classroom. The question of whether I am more than a few billion neurons wired in parallel to resemble an intelligent machine is open to others to answer. From the " inside" the mind says *'I think about* such matters *therefore I am* able to use AI research for building tutoring systems that can reason, argue, problem solve and communicate knowledge. Kelly (1988) concludes his survey on intelligent machines with a statement that I would endorse. He says "... the argument is not that machine intelligence is impossible but that developments to date do not support optimistic projections...".

REFERENCES

Bitzer, D L (1976) The wide world of computer based education, in, Rubinoff, M & Yovits (Eds) Advances in Computers 15, Academic Press, New York.

Borning, A (1981) The programming language aspects of Thinglab, a constraint-oriented simulation laboratory, ACM Transactions on Programming Languages and Systems, 3, 353-387.

Brown, J S et al (1975) SOPHIE: a step towards a reactive learning environment, International Journal of Man-Machine Studies, 7, 675-696.

Brown, J S et al (1982) Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, III, in, Sleeman D and Brown, J S (Eds) Intelligent Tutoring, Systems, Academic Press.

Burton, R R (1982) Diagnosing bugs in a simple procedural skill, in Sleeman D and Brown, J S (Eds) Intelligent Tutoring, Systems, Academic Press.

Carbonnel, J R (1970) AI in CAI: an artificial intelligence approach to computer-assisted learning, IEEE Transactions in Man-Machine Systems, 11,4,190-202.

Clancey, W J (1979) Tutoring rules for guiding a case method dialogue, International Journal of Man-Machine Studies, 11, 25-49.

Collins, A (1977) Processes in acquiring knowledge, in, Anderson, R C et al (Eds) Schooling and the Acquisition of Knowledge, Lawrence Erlbaum, New Jersey.

Dreyfus, H L (1972) What Computers Can't Do: a critique of artificial reason, Harper & Row, New York.

Gagné, R (1965) The Conditions of Learning, Holt Rinehart and Winston.

Gaines and Boose, J H (1988) Knowledge Acquisition for Knowledge Based Systems, in, Gaines B and Boose, J H (Eds) Knowledge Acquisition for Knowledge Based Systems, Academic Press.

Goldberg, A (1977) Smalltalk in the classroom, Report Number SSL 77-2 Xerox PARC.

Johnson-Laird, P. N. (1983). Mental models, Cambridge University Press, Cambridge.

Kelley, J (1988) Intelligent Machines: what chance?,in, Hallam, J and Mellish, C (Eds) Advances in Artificial Intelligence, Wiley.

McAleese, R. (1987). The graphical representation of knowledge as an interface to knowledge based systems, in Bullinger, H.J. and Shackel, B. (Eds) Human Computer Interaction, North Holland.

McAleese, R (Ed) (1989) Hypertext: theory into practice, Blackwell Scientific.

Mitre Corporation, (1974) An Overview of the TICCIT Program, Report M76-44, Mitre Corporation, Washington.

Mitchell D and Dalkir, K (1986) C/CASTE: an artificial intelligence based computer-aided Learning system, in, Computer Assisted Learning- the next five years, Proceedings of the 5th Canadian Symposium on Instructional Technology, Ottawa, 5-7 May 1986.

Murray, D (1988) User Models: a report, ALVEY/KIM Doc 23, University of Aberdeen.

Nelson, T. (1981). Literary Machines. San Antonio, Texas.

O'Shea, T and Self, J (1983) Learning and Teaching with Computers: artificial intelligence in education, Harvester Press.

Papert, S (1980) Mindstorms: Children, Computers and Powerful Ideas, Basic Books, New York.

Postlethwait, S N (1988) Individualisation of Instruction, in, Unwin, D and McAleese, R (Eds) Enclyclopaedia of Educational Media Communications and Technology, Greenwood Press.

Romiszowski, A (1983) Designing Instructional Systems, Kogan Page, London.

Shortliffe, E H (1976) Computer Based Medical Consultation: MYCIN, Elsivier, New York.

Self, J (1986) Artificial Intelligence - its potential in education and training, in Computer Assisted Learning - the next five years, Proceedings of the 5th Canadian Symposium on Instructional Technology, Ottawa, 5-7 May 1986.

Sleeman, D (1984) Basic Algebra Revisited: a study with 14 year olds, International Journal of Man-Machine Studies, 22,2,127-150.

Winograd, T (1972) Understanding Natural Language, Academic Press, New York.

Winograd, T and Flores F (1986) Understanding Computers and Cognition, Ablex.

TECHNOLOGY TRANSFER: THE NATIONAL IMPERATIVE

Wm. M. Coderre
National Research Council Canada
Ottawa, ON

Technology is being recognized increasingly by managers and economists in both the private and public sector as one of the key economic factors in the health of a national economy.

Technology and its transfer is therefore being treated as a critical component in industrial strategy around the world.

In this talk, I will present various aspects of technology transfer as it is being implemented around the world, with particular emphasis on the programs and mechanisms put into place by various governments to facilitate and accelerate the flow of technology.

The past, current and coming policies of the Canadian government in particular will be presented and discussed.

LE TRANSFERT DE LA TECHNOLOGIE: L'IMPÉRATIF NATIONAL

Les gestionnaires et les économistes, aussi bien du secteur privé que du secteur public, reconnaissent de plus en plus que la technologie est l'un des principaux facteurs économiques sur lequel repose la bonne santé de l'économie d'un pays.

La technologie et son transfert apparaissent désormais comme une composante essentielle de la stratégie industrielle partout dans le monde.

Dans cette communication, le conférencier exposera les divers aspects du transfert de technologie, indiquera comment on le met en oeuvre dans les différents pays du monde en soulignant surtout les programmes et les mécanismes mis en place par les différents gouvernements pour accélérer la circulation des connaissances technologiques.

Plus particulièrement, seront abordées et discutées les politiques passées, présentes et futures du Gouvernement canadien.

TECHNOLOGY TRANSFER: the National Imperative

Dr. Wm.M. Coderre

National Research Council Canada

We Canadians are facing one of the greatest challenges we have ever faced as a nation. International competition is threatening the standard living we have come to regard almost as a birthright. The problem is, right up to the present, we have been living (as it were) off the fat of the land. Our national wealth is traceable for the most part to the resources which a bountiful land, sea and forest have provided. Today, however, it has become increasingly apparent that a "Land does not live by Bread-basket alone."

The new commercial advantage determining the wealth of nations is not in what one can grow, or catch, or dig up, or cut down; no, the wealth of nations to an unprecendented degree is now based upon the effectiveness with which they manage technology and innovation.

You are here at this conference to discuss and exchange information on how modern technology is transforming the field of education. Not one among you is unaware that new computer-aided techniques available today, and even more-so in the future, will provide opportunities for improvements in the efficacy of the didactic process. This technology-driven revolution is not just found in schools or in factories, or in industrial offices. The neighbourhood pizzeria is now taking orders by facsimile machine and is delivering the pizza while it is being cooked in a mobile kitchen.

Perhaps this is a trivial example of the invasion of technology where it was not anticipated, but I use it to illustrate the pace and pervasiveness of the changes which technological opportunities afford.

Is Canada well placed to respond to the technological challenge? Are we destined to be one of the winners or one of the losers in this new form of economic warfare? I sincerely wish I could say for certain that we will use our many advantages to forge to the front of the pack, but frankly, the outcome of this race is anything but certain at this time.

Situation Analysis

Let us look at some of the vital statistics comparing Canada to her competitor nations: The facts indicate that on every critical technical measure, Canada is lagging behind. We spend half as much of our gross domestic product on R&D, and when we look at our R&D budget of industrial firms, the figures are twice as bad. We produce and employ only half as many engineers and scientists as do our major trading partners. In fact, as you educators well know, the prognosis for any increase in the number of technically qualified Canadians is not good. Not only is the percentage of students opting for the maths and science courses, which are essential, low in Canada, but a decline in the absolute number of students is also foreseen.

The balance of trade figures for manufactured goods, especially for those with high levels of technology, show the result of this national shortfall in the necessary investments to keep Canada competitive. In every manufacturing category, Canada imports more than it exports. Our raw materials and other primary products are sustaining our economy but for how long? The technological revolution is a world wide phenomenon. Paper making factories have been designed and built in Japan and floated up the Amazon where this up to the minute plant can effectively compete with Canadian paper companies who now have to harvest trees further and further up the mountain side or further and further from the river. Centuries of near neglect in reforestration are coming back to haunt us, while the Swedes and the Finns are already harvesting forty year old trees from carefully cultivated woodlots.

Canada has always prided itself as being the bread-basket of the world, but the technological breakthroughs of the so-called Green Revolution have improved crop production around the world to such an extent that the only major factors for significant overseas sales are natural disasters and droughts; hardly a reliable base upon which to build a stable business. The mineral wealth of Canada continues to be a source of strength, as does the fishery, but new engineered industrial materials have mining companies diversifying as their markets are eroded, and technical breakthroughs in the fishery, not to mention in

aquaculture, are already transforming this hitherfore staid industry.

Canadians still have a lot going for them. We have a very well educated population, one of the most stable political environments in the world, an abundance of energy and raw materials, a surprisingly strong entrepreneurial success record, in fact, everything any country would wish to have to succeed in world competition. On top of this, we have just concluded a free trade agreement with the world's biggest and most open market, the USA.

The Technological Revolution

What can we do? What should we do to assure a good standard of living for our children? One dimension of what we should do has already been demonstrated to us by the Japanese over the past few decades. They have become masters at quickly gathering knowledge from around the world and translating it into know-how at work in Japanese factories, producing high quality, high value products for sale around the world. We in Canada are going to have to learn to be just as efficient as the Japanese in importing, transforming and exploiting technology if we are to be home to competitive firms and factories in the next century.

Technology is no longer the preserve of the few. Technical breakthroughs which will change the marketplace are springing up all over the world and are quickly being disseminated to all other parts of the world. A recent example of this was the publishing in Zurich Switzerland of the discovery of a new class of superconducting material which promised to allow zero resistance transmission of electricity at temperatures much higher than previously attainable. Before six months had passed, the work of the IBM Zurich team had been duplicated and surpassed in the US, in Japan, in Germany, in fact, all over the world. Back in the sixties, we used to talk about the need to have "Technological Sovereignty" in Canada. It was believed that every sovereign state had to be able to possess and to control all of the critical and strategic technologies that its economy would require in order to be secure against denial of access by a competitor. The trouble with this notion is that technology no longer knows any political boundaries. Trying to maintain technological sovereignty is like trying to nail jello to the wall. And even if you succeeded in keeping your discoveries in, you would soon find that your prize has been made obsolete by the new discovery somewhere else in the world. The US for example, was the inventor and developer of the integrated circuit yet today, the bulk of such circuits are being sold by Japanese companies. The laser was a British invention, yet the most successful exploiter to date are Americans.

Today, the critical talent needed for success is to be able to recognize the best and most appropriate technology from around the world, including, where appropriate, from the university across the street, or the government laboratory in the next province and to apply it in an innovative way in products or services with world-wide demand. Easy to say, not so easy to do.

Three Classes of Technology Transfer

This process is known as technology transfer. Aggressive entrepreneurs in all countries have recognized the new reality, that it is no longer enough to carry out research and development in your own laboratories. You must be ever alert to changing technical opportunities and technology-based threats to your markets. Never has it been so true that the surest way to go out of business is to keep on doing what worked yesterday.

There is more than one form of technology transfer. Different methods are employed according to the circumstances of the firm. Within the wide spectrum, we can identify three general classes of technology transfer: Imitative, Adaptive, Innovative.

Imitative Technology Transfer

Imitative technology transfer is that process whereby a firm acquires the right to manufacture someone else's products usually for the local market. This is achieved through the licencing of the rights to manufacture the product, and through acquiring the design, engineering drawings or recipes necessary to duplicate the originator's work. Imitative technology transfer also includes the purchasing of processing technology to improve the production of an existing product. At times, technology transfer can be simply buying-in other companies' products to use in your manufacturing systems. When all this is accomplished, what we have in Canada is a copy - a duplication - an imitation of that which exists elsewhere. It can be an effective, low cost, strategy but it has serious drawbacks.

First, the technology, although proven, and hence relatively risk free, is already on its way to obsolescence before you obtain it. Furthermore, the first users have a good lead down the learning curve, so your profit margins in an open economy will be less. Finally, since you are merely a user of the new technology, you are not in a good position to improve on it or adopt it, so when it is obsolete, you will have no choice but to shop around for another newer technology injection if you want to stay in business.

This form of technology transfer has been typical of the flow of technology from developed nations to underdeveloped countries. It has the effect of allowing the originating firm to receive extra income from its established technology, while assuring that potential competitors stay behind them. The licencee has been granted a piece of a shrinking pie.

Adaptative Technology

The second form of technology transfer is called Adaptive. In this case, the recipient firm has an engineering team of its own who can comprehend the basic principles of the new technology and can transform it to better suit local conditions, or to use local, more available or perhaps less expensive, materials. Sometimes, the adaptation can open new markets and applications. If this is done well, the adaptation can be a significant improvement on the original. When this happens, there can also be a synergistic and strategic partnership between the two firms. They share each-other's know-how to create larger, more diversified markets for both. In effect, together they bake a bigger pie with a bigger piece for both.

Innovative Technology Transfer

The third form of technology transfer is the riskiest and most demanding on the company, but it also promises the greatest rewards for those who succeed. Innovative technology transfer has most of the benefits of original invention and in some ways is even better. In this case, a firm acquires rights for the exploitation of an intellectual property which is, as yet, unproven in the marketplace. Such technology abounds in universities and research laboratories around the world. If you select wisely and at the right time, it is possible to be first in extremely lucrative markets, without having had to carry all the risks and expenses inherent in establishing and maintaining your own research program. But, if this strategy is to be successful, how is the firm to know where the appropriate new technology is to be found, and whether it is as good as its inventors and developers think it is? When this strategy works well, it is usually into a firm with a high degree of technological competence under its own control, or one which can avail itself of solid good advice.

I have been arguing that technology transfer is essential for Canada's economic growth and well-being, but is it really necessary to change anything? Haven't Canadian companies, particularly multinational subsidiaries been doing this for decades?

Impact of Free Trade and Global Markets

It is a well known fact that much of the industrial infrastructure of Canada is under the control of multinational corporations. An analysis of their spending shows that branch plants do not spend as much on R&D as do domestic Canadian firms. Perhaps this is not a problem. Don't these branch plants have ready access to their central corporate research facilities? Yes they do, and technology does flow into Canada through this route, but, without local R&D teams and the local innovative creativity needed to develop world class new products, this remains no more than imitative technology transfer, with sales anticipated only in the limited domestic market. Frankly, this approach will not be adequate in today's competitive marketplace.

In fact, with free trade a reality, there is less and less reason for a branch plant to service a limited domestic market. The firms which will thrive in Canada, and in fact, those who are already succeeding, are those whose markets are found all around the world, or at least with a significant penetration of the nearest, easiest export market, the USA.

In a recent article in Computing Canada, the newly appointed Minister for the new Ministry of Industry, Science and Technology, Harvie André, was quoted as saying that Canada must develop its own technology to retain and improve the competitive edge of its industry and business, both domestically and internationally. He identified the branch plant economy as a particular problem. "If three or four branch plants close due to the free trade and the economic restructuring we are witnessing" he said, "that will, perhaps, facilitate the creation of one Canadian firm to fill the void, one that will conduct R&D. Branch plants just do not do research".

It is perhaps encouraging to note that both Mr. André and the new Minister of State (Science and Technology), Bill Winegard, have doctorates in engineering, bringing the most qualified incumbents ever into these two increasingly important portfolios. This is just the latest move on the part of the government to improve the environment for technology in the Canadian economy. In the past few years, the government has developed a Federal-Provincial accord on Science and Technology, it has increased federal support in certain strategic technologies, it has increased the budget to the very effective NRC IRAP program and to the External Affairs, Technology Inflow Program. The Prime Minister himself chairs the National Advisory Board on Science and Technology.

The competition, however, is not standing still. The Japanese are pushing to increase their national investment in S&T to above 3%. The Europeans have created major programs like EUREKA, SPRITE and RACE, the European Space Agency and other initiatives to encourage pan-European cooperation in technology-based projects.

In the UK, new programs have been created to encourage the industrial exploitation of both military and civilian S&T from government facilities. Governments in all parts of the world are improving the environment and the infrastructure for industrial innovation.

Support Mechanisms for Technology Transfer

What help can a Canadian entrepreneur find if it wishes to take up the competitive challenge to find and use appropriate new technologies to support his innovation? The news is good, and getting better.

First, let us look at technology transfer mechanisms already in place in Canada.

Although the Canadian industrial R&D effort is one of the weakest in the Western World, Canada still has a legacy of relatively strong public sector science and technology establishments. Through the work of the National Research Council and since 1978, the Medical Research Council and the Natural Sciences and Engineering Research Council with their excellent process for peer-adjudicated grants, the university research base in Canada is still quite strong. More recent, NSERC initiatives have encouraged university-industry liaison through strategic grants and university-industry projects. The latest in these initiatives is the current process identifying the best of several hundred proposals for Network Centres of Excellence which will link the best university research teams across the country in partnership with industry and federal and provincial research centres. (As an aside, I am proud to note that NRC was invited to be a participant in no fewer than 52 of the 153 submitted proposals).

To increase even further the ability of universities to transfer their expertise to the private sector, many provinces have supported the establishment of technology transfer, or innovation offices within the universities. NRC, through its IRAP program, has also contributed funds to various universities to encourage the setting up of these important interface agents. IRAP also provides the infrastructure to network all technology transfer agents across Canada, to better able them to exchange information on opportunities and mechanisms for improving their operations.

The National Research Council

At the federal level, the strongest support for technology transfer comes through the National Research Council. The NRC operates three major activities, each contributing in its own way to helping industry to acquire and to make effective use of technology.

NRC Laboratories and Facilities

First, there are the national research laboratories and major facilities of the NRC. Fifteen laboratory installations are found in seven different provinces. In St. John's, the Institute for Marine Dynamics is a national centre of excellence in all aspects of vessels, platforms, and shore facilities operated in the conditions found off Canada's four coastlines. Here in Halifax, the Atlantic Research Laboratory maintains expert programs in marine biochemistry and advanced ceramics. It was here that, working day and night, NRC scientists cracked the mystery of the deadly mussel toxin and saved the shellfish industry of the east coast.

In Québec NRC operates the Biotechnology Research Institute and the Industrial Materials Research Institute, both at Montréal, and has contributed to the establishment of the National Optics Institute in Québec City.

Starting from the West Coast, we find in B.C. the Dominion Astronomical Observatory and the Dominion Radioastronomical Observatory. In fact, NRC's western reach extends to Hawaii where we are co-operators of two major telescopes on top of the volcano, Mauna Kea.

In Saskatoon, the Plant Biotechnology Institute is developing valuable techniques aimed at improving the silvaculture and agricultural industry of Canada. In Winnipeg, the Canada Institute for Industrial Technology is currently working in partnership with 31 different firms on projects of their design as they share space in NRC's laboratory.

Finally, in Ottawa, NRC houses its Divisions of Physics, Chemistry and Biology, the National Aeronautical Establishment, the Herzberg Institute of Astrophysics the Institute for Research in Construction, the Divisions of Electrical and Mechanical Engineering and the Division of Informatics.

In all of these excellent facilities, NRC scientists and engineers are working with and for Canadian firms to help them improve their competitiveness through the application of appropriate technology.

Firms are invited to send their engineers into NRC labs to familiarize themselves with the latest, best techniques, or to use state-of-the-art facilities and pilot plants. NRC's scientists and engineers travel the world, often at the invitation of major research organizations, who are their colleagues, seeking out and evaluating new breakthroughs which may have good Canadian applications. Firms join NRC in cooperative projects and consortia to work on significant new technologies of common interest.

For instance, the seven Canadian manufacturers of HiFi speakers have joined NRC's Physics Division in a program aimed at creating the smart adaptive speaker before the Japanese do it and take away their hard won customer base.

Canada Institute for Scientific and Technical Information (CISTI)

The second major element of NRC is CISTI, the Canadian Institute for Scientific and Technical Information. CISTI is one of the most modern and effective information providers in the world. From its collection of more than 4 million titles and its 42 electronic databases, CISTI delivers more than 400 thousand document requests each year. Its clients can have instant access through on-line inquiry and automatic request services.

Through CISTI, firms in Canada can and do track significant changes in their fields of interest. In fact, CISTI offers a special monitoring service to clients where CISTI sends out regular updates according to a client's interest profile.

Industrial Research Assistance Program (IRAP)

Finally, NRC offers the Industrial Research Assistance Program (IRAP). Through IRAP's 250 Industrial Technology Advisors and Project Managers, working from 150 offices in 75 different agencies across Canada, 5,000 Canadian firms are helped to find, to acquire and to learn to make effective use of appropriate new technologies. IRAP's contribution budget of almost $80 M supports technology transfer from federal government departments, provincial research institutes and university laboratories to Canadian firms. In cooperation with the department of External Affairs, IRAP offers the services of the Technology Inflow Program where firms are helped to go abroad to check out technical opportunities ferretted out by the Science and Technology Counsellors and Technology Development Officers stationed in all of the major countries of the world.

Technology is the smart way to get ahead and to keep ahead in a rapidly changing, technology dominated marketplace.

We in Canada have a marvelous opportunity to take our natural advantages, add to them our entrepreneurial skills and our well educated people and move to the front ranks of the modern economic states. The federal government has created support infrastructures to help firms to help themselves through technology transfer. It is vital to all of us who are proud to call ourselves Canadians that we rise to the challenge, take control of our own destiny, learn to make effective use of technology transfer opportunities and prosper.

ELEMENTS OF A PROVINCIAL EDUCATIONAL TECHNOLOGY INTEGRATION PLAN

J. Travers
Curriculum Support Branch
Alberta Education
Edmonton, AB

Decisions on the application of technology in education must be based on those areas that exhibit both educational need and represent the current capabilities of the technology. All too often, technological implementations in education have focussed on glitzy applications or on areas that did not, in most circumstances, address high educational needs (e.g., drill and practice language arts software). Once ascertained, the target application areas must be integrated into the curriculum rather than remaining adjunctive. We must, though, go further than integration with the curriculum. Teachers and students must also be given access to the learning materials and leadership in their application in the classroom. Then maybe, just maybe, technology will begin to have a major impact on learning and "pay its way" in the educational enterprise.

LES ÉLÉMENTS D'UN PLAN PROVINCIAL D'INTÉGRATION DE LA TECHNOLOGIE PÉDAGOGIQUE

La décision de recourir à des outils technologiques en éducation doit reposer à la fois sur un besoin pédagogique et sur les capacités actuelles de la technique. Trop souvent, on a mis en oeuvre en éducation des solutions technologiques essentiellement spectaculaires ou destinées à des disciplines qui, la plupart du temps, n'exigeaient pas une démarche pédagogique élaborée (par ex. apprentissage et mise en pratique d'un logiciel de langage des arts). Les solutions visées, une fois identifiées, doivent être intégrées au programme de cours et non être un simple complément. Cependant, le travail ne doit pas se résumer simplement à les intégrer au programme, il faut mettre à la disposition des professeurs et des étudiants les outils pédagogiques et les conseillers qui assureront leur mise en oeuvre dans les classes. C'est à ce prix, et seulement à ce prix, que la technologie pourra avoir une incidence marquante sur l'enseignement et s'imposer dans le marché de l'éducation.

Elements of A Provincial Educational Technology Integration Plan

by
John Travers
Associate Director
Technology and Resource Integration Unit
Alberta Education
Edmonton, Alberta
Canada
T5K 0L2

Preamble

The best possible education for all students. This overriding goal must underlie any plan for the integration of technology into education. That is, technology in education is not primarily an end in itself, but it is a means of achieving the higher educational goal of *improved student learning*. A major challenge is, then, how best to achieve improved student learning through the use of technology.

If the past is any guide, the payoff of the utilization of technology in education has often been less than early expectations. The use of technology in schools has primarily been **adjunctive** rather than integrated into the main activity of instruction and learning. Clearly, if provincial resources are to continue to be directed to this area, technology must be directed towards appropriate and effective learning situations.

This paper describes some of the planning currently occurring within the province of Alberta that will result in a provincial educational technology plan.

Background

Attempts to integrate technology have a long tradition in the educational system. Radio, film, educational television and Computer Assisted Instruction (CAI) are a number of examples of technological innovations that have attempted to improve student learning. With the advent of the newer technologies, such as microcomputers, interactive satellite, artificial intelligence and CD-I, education continues to be encouraged to embrace new technological solutions. Dissidents to these trends might contend that previous attempts to implement technology have not been particularly successful in the educational milieu and that the newer technologies will follow the example of their predecessors. This may be the case, but a number of differences exist today that imply that educational technology has considerable potential in selected areas.

Trends in Computer Technology

Although technology is constantly changing and difficult to predict, the rapid change in the capabilities of computers and in the computing

industry is no less than astounding. For example, in less than 10 years the personal microcomputer has gone from a "gadget," assembled by electronic enthusiasts, to an industry that now threatens the balance sheet of some major computer companies. Both processor power and computer memory storage have increased by factors of hundreds and thousands of times over this short period. These hardware changes are having an appreciable effect on the quality of software and the *ease of use* of the current generation of equipment. The result of these changes is that the computer has rapidly become a productivity tool for many industries, and education stands poised to make similar gains.

For example, a word processor can be used to facilitate the development of student writing abilities, newer optical storage databases may facilitate more informative and educational access to information, Computer Managed Learning (CML) systems promise to decrease the onerous marking loads of teachers in continuous learning situations and current authoring systems promise far easier and more efficient production of CAI software. Also, new interactive random access storage devices such as CD-I, coupled with computers, have the potential to add significantly to the individualization of student learning.

Partly as a consequence of significant oil revenues over the past 30 years, Alberta has been on the forefront of the use of computers in education. One of the most evident examples of this activity was the decision by the province to establish an initiative for microcomputers in schools in the early 1980's. Since that time Alberta has acquired over 27,000 computers for schools, with an overall student computer ratio of less than 17:1 and a number of the high schools operating with less than 8:1 ratios.

One approach to describing the application of technology in education is to use the application definitions "learning about, learning using, and learning through" technology. "Learning about" technology refers to consideration of the technology as the *object* of investigation and might include courses of study such as Computer Studies, a Science and Technology course or the study of specific technologies within a course (e.g., electron microscopes, mass spectrometers, etc.). "Learning using" the technology applies primarily to the use of technology as a tool (e.g., using a word processor to improve writing abilities). "Learning through" the technology applies mainly to the use of technology as an instructional or delivery device (e.g., Computer Assisted Instruction [CAI]).

Initial applications of the microcomputers in the province were to *learn about* them or treat them as objects of instruction, and computer literacy courses were the result. Another early application of these computers was to attempt to use them to provide instruction. Alberta Education established a Courseware Clearinghouse to evaluate and recommend authorization of software for schools. Over a five year period, the Clearinghouse evaluated over 2000 software programs and authorized approximately 200 programs, most of

these at a supplementary learning resource level.

However, a dominant question remains. Has the expenditure of more than $63M for computers in the province resulted in improved student learning?

A Vision for Technology Integration

A goal state for technology integration might be as follows:

Goal State:

Appropriately applied and implemented, the integration of technology across the curriculum will result in improved student learning and the best possible education for all students.

Technology integration is then defined as the linkage of technology with the curriculum to address educational needs. In this way, future applications of technology within education will be integral and not adjunctive to the educational process.

Assumptions

A number of assumptions undergird the directions presented in this paper.

- *That technology can improve student learning if it is targetted towards educational need areas.* Technology, if it is to make a difference, should be directed to recognized educational needs (e.g., a word processor can be a successful tool in the development of student writing abilities, a Computer Managed Learning (CML) system can assist the teacher in the marking and tracking of students).

- *That educational applications must use technology in ways that reflect its current capabilities.* Each technology has particular strengths and weaknesses that relate to characteristics of the technology and its current stage of development.

- *The implementation strategy for technology integration in education is to target situations where technology both addresses educational curricular needs and is appropriately applied to areas that reflect its capabilities.*

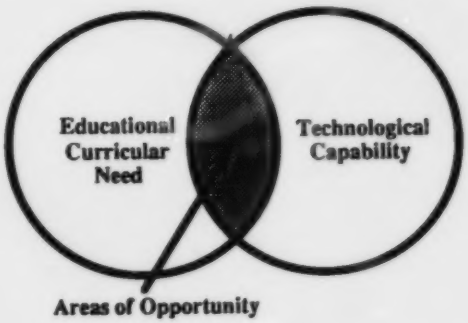


Fig. 1 Educational Curricular Needs and Technological Capabilities

- *That technology must be integrated with the curriculum to have major effect.* Rather than being adjunctive, the integration of a particular technology must significantly contribute to a learner expectation of the curriculum. In most cases this integration should occur in the early design stages of curriculum development or redevelopment.

- *That technology integration will have major effect when teachers have ready access to instructional technology materials which directly assist in the teaching process and to student learning resources which students use under their direction.*

- *Effective use of technology for student learning requires attention to: (a) "the hardware" requirements (e.g., computers, etc.); (b) the learning resources (e.g., software, etc.); and, (c) the knowledge of how to use the technology either as an instructional device used by the teacher or as a learning tool used by the student.*

Implementation Strategies

Planning can then focus on three major areas pertaining to the integration of technology into the curriculum. These are:

1. Integration of Technology into Curricula

Integration of technology into the curriculum is a necessary first step. In the past, curriculum development often proceeded independently from attempts to implement technology and resulted in adjunctive technology applications in education. Integration into the curriculum must include establishing general and specific technology integration learner expectations within the curricular Programs of Studies and the targeting of applications of technology within specific curricular areas.

2. Teacher Support Materials and Student Learning Resources

Teacher Support Materials

In the early stages of a particular technology's implementation it may be necessary to produce support monographs to assist teachers in the implementation of technology in specific application areas. One example of this direction in Alberta was an initiative entitled, "The Writing Process Using the Word Processor". This project involved the linkage of the writing process and the use of the word processor as a tool to improve student writing abilities. A teacher leader manual was developed which detailed the use of the word processor in developing student writing abilities. To facilitate the integration of the word processor into the curriculum, learner expectations in the Language Arts area have been drafted.

Student Learning Resources

As part of the "Writing Process Using the Word Processor" initiative described above, an extensive search was conducted for word processing packages. From an initial identification of more than 30 software packages, six word processing packages reached the final selection stage. Quotations on provincial licenses for these packages were received from the vendors and resulted in two word processing packages being acquired for provincial distribution.

Another example of integrated student learning resources relates to one of the more difficult problems associated with the provision of equitable educational opportunities in the rural areas of the province. An example of this problem has been the difficulty in offering senior high school level courses to very small numbers of students in small rural schools. Mathematics 30 (grade 12) was chosen as an area that presented opportunities for the use of technology in the provision of equitable educational opportunities via computer assisted instruction. Of particular importance was the choice of preparation of this material using a newer generation authoring system.

Distance learning materials have been incorporated into the system with levels of enhancement including dynamic graphics and student performance recordings. The goal is that these materials in conjunction with the distance learning resources, will provide effective individualized instruction and continuous learning for students in the small rural schools. Evaluation of this initiative should indicate if future courseware development initiatives are warranted.

The province is also very active in the use of Computer Managed Learning applications for addressing the learning needs of students and teacher support needs in these rural situations. Currently, this approach is being used in nine high school mathematics courses in rural schools throughout the province.

3. Planning and Research

A survey entitled "Microcomputers in Alberta Schools 1986" revealed a wealth of information on the extent of computer implementation and problems associated with the use of computers in schools. This survey has proved to be foundational to many of the initiatives undertaken during the past two years in the province. A similar survey will be conducted again in 1989/90 to determine the status and use of computer resources currently in Alberta schools.

Information will be sought on:

- use of resources acquired through licensing initiatives
- type, variety and uses of computer software in schools
- type, variety and uses of computer hardware in schools
- major target areas for technology integration
- identification of major blockages to technology integration

This information will be foundational to a technology planning process proposed for the fall of 1989 to chart future directions in this area on a provincial scale. Information and recommendations from the survey and planning process will be used as input for an Educational Technology Plan scheduled for completion by early 1990.

Critical Issues in Technology Integration

While the intent of provincial planning is to be as comprehensive as possible, a number of issues can be identified that require resolution.

These include:

- The need to develop effective approaches for evaluating and integrating courseware in newer dense media. In the past, courseware reviews have focussed on the analysis of programs representing small segments of a course, but with storage technologies available today, entire courses will be available on a disk. How will provincial agencies determine the curriculum fit of these resources and integrate these resources with the unique curricular and pedagogical approaches that exist in each jurisdiction?

- The need to find ways to integrate and take advantage of the technology available to children in the home. Increasingly, the computer will become a tool required by each student and a part of their own school materials. How and when will the transfer of responsibilites from provincial acquisitions to student-owned computers occur?

- Related to the previous issue, every jurisdiction is faced with the need to upgrade the current computer base in schools. At present rates, new computer generations appear every three to five years yet schools will continue to use these resources for more than ten years. This implies a "steady state" of three generations of computers within the school at any one time.

- There is increasing emphasis in the curriculum on the development of problem solving and thinking skills by students. The development of information-accessing skills by students and the linkage of these skills to the use of data bases constitutes a major future area of focus.

- The need to articulate activites with the Faculties of Education to ensure that new directions in technology integration are reflected in teacher preservice training. Failure to do this will leave the field with a continuous inservice challenge.

Summary

Alberta has an unique opportunity to lead in the use of technology in schools. The province has a large and well-established base of computers. A curriculum review of secondary education programs creates the opportunity for the curricular integration of technology at the design stage and will also facilitate the inclusion of technological resources in support materials for the schools. This confluence of activity creates a fertile opportunity for the integration of technology into the curriculum. If successful, the use of technological resources within schools should become a common-place experience for students and, if the targets are well-chosen and implemented, technology will make a significant contribution in the provision of the best possible education for all students.

ARTIFICIAL INTELLIGENCE/

INTELLIGENCE ARTIFICIELLE

AN INDUCTIVE LEARNING PROCEDURE FOR ADAPTIVE INSTRUCTIONAL SYSTEMS

J. Nesbit
Institute of Information Sciences and Electronics
University of Tsukuba
Tsukuba, Japan

S. Hunka
University of Alberta
Edmonton, Ab

Gagne-style learning hierarchies, which are often used by instructional designers to represent the prerequisite structure of course objectives, are a practical way of organizing adaptive instructional systems which deal with global course structure. When a student exhibits difficulty in learning an objective, the learning hierarchy serves a diagnostic function by suggesting for testing (and possible review) one or more prerequisite objectives which may have been forgotten or never mastered. This paper describes and evaluates a new method, the inductive learning procedure (ILP), which uses machine learning techniques to improve on this diagnostic function of learning hierarchies. The major premise of the project reported here is that prior information about the student, in the form of an attribute vector, can be used by a set of decision rules to reduce the amount of testing required to determine which preprequisites are missing. Because it is rather difficult for authors to precisely specify such decision rules in advance, ILP has been designed to either refine the roughly hewn rules provided by an author, or learn rules from scratch if none are provided. ILP dynamically builds and modifies these decision rules as it encounters new students, basing its learning on observations of whether these students pass or fail tests measuring the mastery of the preprequisite objectives. An evaluation of the performance of ILP with artificial student models showed that, under certain plausible conditions, the decision rules it learns will greatly reduce the amount of testing required to determine which prerequisites are not mastered. Guidelines for using the procedure and areas in need of improvement are also indicated.

UNE MÉTHODE D'APPRENTISSAGE INDUCTIVE POUR LES SYSTÈMES D'ENSEIGNEMENT SOUPLES

Les hiérarchies d'apprentissage de style Gagne, souvent utilisées par les concepteurs de programmes d'enseignement pour représenter la structure préalable des objectifs des cours, constituent une façon pratique d'organiser des systèmes d'enseignement souples portant sur la structure globale des cours. Lorsqu'un étudiant éprouve de la difficulté à atteindre un objectif, la hiérarchie d'apprentissage permet de faire un diagnostic et propose de tester (et peut-être de revoir) un ou plusieurs des objectifs préalables que l'étudiant peut avoir oubliés ou n'avoir jamais maîtrisés. Ce document décrit et évalue une nouvelle technique appelée méthode d'apprentissage inductif (MAI); cette dernière utilise les techniques d'enseignement assisté par ordinateur pour améliorer la fonction de diagnostic des hiérarchies d'apprentissage. Le premier principe du projet à l'étude ici est qu'il soit possible d'utiliser des données sur l'étudiant, sous la forme d'un vecteur d'attributs, selon un ensemble de règles de décision afin de diminuer la quantité de tests permettant de déterminer les conditions préalables manquantes. Comme il est plutôt difficile pour les auteurs de préciser ces règles de décision à l'avance, la MAI a été mise au point pour raffiner les règles brutes émises par un auteur ou pour apprendre les règles à partir du point de départ si aucune règle n'est fournie. La MAI élabore ces règles de décision et les adapte de façon dynamique en fonction des étudiants, en faisant des observations sur le succès ou l'échec des étudiants qui passent les tests de mesure de la maîtrise des objectifs fondamentaux. Une évaluation de la performance de la MAI appliquée à des sujets artificiels a démontré que, dans des conditions vraisemblables, les règles de décision qu'elle apprend réduiront considérablement le nombre de tests à effectuer pour déterminer les éléments préalables que les étudiants ne possèdent pas bien. Les principes directeurs de l'utilisation de cette méthode et les secteurs à améliorer sont également indiqués.

AN INDUCTIVE LEARNING PROCEDURE FOR ADAPTIVE INSTRUCTIONAL SYSTEMS

J. C. Nesbit, S. Hunka*

University of Tsukuba, *University of Alberta

One can imagine machine learning being incorporated in all three components of Hartley's (1973) framework for adaptive instruction. The expert model (or task model) can acquire improved heuristics by analysing its own performance, as in LEX (Mitchell, Utgoff, & Banerji, 1983), or by recognizing when a student has obtained a better solution to a problem than the one it has provided, as in Kimball's self-improving tutor (Kimball, 1982). Student models which incorporate learning permit different instructional strategies to be tried on the model so the best can be selected for presentation to the student.

A topic which has received little attention, and one with which this paper is concerned, is the use of inductive learning methods to improve diagnostic decision rules in the teacher model. In this approach, student models are examples which an inductive learning procedure uses to refine rules initially entered by a course author. The hope is that the eventual performance of such a teacher model will parallel that of an experienced teacher who, upon encountering a student with some problem or other, applies rules learned from experience with previous students. Unlike the problem of incorporating learning in student models, there is no need to be constrained to reproducing processes specific to human learning.

Gagne-style learning hierarchies have primarily served as a guide to the sequencing of instructional objectives, but they can also provide a valuable diagnostic function. When a student exhibits difficulty in learning an objective, the learning hierarchy points to prerequisite objectives which may have been forgotten or never mastered (Heines & O'Shea, 1985). In this case the instructional system can either review all the prerequisites, or it can be more adaptive by identifying which of the implicated prerequisites are actually in a state of non-mastery. The cost of the adaptive strategy is time and tedium of the testing required to determine the student's understanding of each of the implicated prerequisites.

The major premise of the project reported here is that prior information about the student can be used by decision rules to reduce the amount of required testing. It is supposed that, at the time a course is being designed, course authors cannot provide an accurate set of such rules because doing so requires one to observe rather large numbers of students progressing through the completed course. Thus, an inductive learning procedure (ILP) is presented here which refines rules given by the author, or learns them from scratch if none are provided.

Due to space limitations imposed on proceedings, this report can be no more than a rather sketchy outline of ILP, especially regarding its experimental evaluation. A more thorough account is available in Nesbit (1988) and also Nesbit & Hunka (1988).

ILP was designed and implemented for use in adaptive instructional systems of the kind based on learning hierarchy representations of the subject matter. It is assumed here that the rules learned by ILP are used to identify prerequisite objectives that a student exhibiting difficulty with an objective (let us call it the prime objective) has forgotten. The outer control procedure, which invokes the inductive learning procedure, takes the following actions whenever such a student is encountered:

1) Recommendation phase. One or more prerequisite objectives are recommended for testing by applying the learned rules to the model of the failing student.
2) Testing phase. The student is tested on the recommended prerequisites. When a test is failed the student is routed to the appropriate unit for instruction. When all such units have been successfully completed the student is returned to the prime objective.
3) Learning phase. The student model and the test results constitute examples which are used by ILP to improve the rules.

There were three fundamental decisions made at an early stage of the project which strongly determined the design of ILP. First, an example is supplied to the learning phase only as a result of a student being explicitly tested on a prerequisite. Indirect evidence for an example is not allowed. For instance, when tests reveal mastery on k-1 out of k prerequisites, the student model is not automatically supplied to ILP as an example of a student in need of the remaining prerequisite. Although the use of such indirect evidence could speed the learning process, it was considered to rely too heavily on the assumption of absolute validity of the learning hierarchy.

Second, the form chosen for the examples was a simple conjunction or list of attribute-value pairs called an attribute vector. The attribute vector contains whatever information about the student the author chooses to include (e.g., scores on aptitude tests, sex, age, objectives mastered, responses,

and so on).

Third, it was decided that rules hypothesized and directly entered by authors should be given high priority as a means of biasing the learning process. It was believed this form of background knowledge offers the best starting point for the search conducted by ILP.

The qualities which characterize this inductive learning task are:
° Incremental rule refinement. Students passing through the testing phase constitute a stream of examples for the learning phase. The learning procedure should try to improve the rules after each new example. If ILP waited until some large set of examples were collected, students represented by this set would not benefit from partially learned rules.
° Shifting of target concepts. Student populations may fluctuate over time necessitating changes in the rules.
° Passive example acquisition. ILP has no control over the order and kind of examples it encounters.
° Disjunctive concepts. ILP must be capable of learning disjunctive concepts because there may be several reasons why a prerequisite is not mastered, and we should expect positive examples to cluster in disjoint regions of the example space. As Bundy et al. (1985) observed, in order to support learning of disjunctive concepts the learning system should keep all examples encountered.
° Noisy examples. The classification of examples, as well as the attribute values comprising examples, will be the result of educational tests and other sources subject to error. Also, there is no guarantee that the set of attributes chosen by the author will provide a sufficient basis for discrimination between positive and negative examples.
° Linear or ordered attributes having large value sets. Authors will want to use ordered attributes such as aptitude scores and response latencies. The need to support ordered attributes eliminates the CLS/ID3 approach (Quinlan, 1983) from consideration for this application.

Lee and Ray's (1986a) probabilistic rule generator (PRG) comes closer than previous procedures to satisfying the requirements of the application considered here because it learns disjunctive concepts, tolerates noisy examples, and recognizes both linear and nominal attributes. Although PRG does not naturally accommodate refinement of rules entered by the user, Lee and Ray (1986b) have provided an indirect mechanism supporting rule refinement by initializing the example set with artificial examples conforming to the user-supplied rule. The disadvantage of this approach is there are often many rules which agree with a given set of examples, and PRG has no way of determining the rule intended by the user. Another disadvantage of PRG is that it does not admit new examples once the learning process is under way. This means that, if it is to be used for incremental learning, PRG must start from scratch after each new example, receiving no benefit from its previous efforts. In spite of these shortcomings PRG should be regarded as a potential alternative to the procedure presented here.

Unlike PRG, ILP starts the search with the set of rules learned after the previous example and tries to improve it by applying generalization or specialization operations. Examples migrate between states of agreement and disagreement with the current rules, as the rules are modified. ILP keeps track of the state of each example and uses disagreeing examples to guide the application of operations.

Attributes

For each node in the learning hierarchy having more than one prerequisite, the author specifies a set of attributes to comprise the attribute vectors used by ILP for building rules for that node. The most common attributes, such as TimeSinceMasteryOfPrereq, would be available from a menu. Others may have to be collected by procedures written by the author.

The author also determines the types of the specified attributes. Following Michalski (1983) three types of attributes would be allowed: nominal, ordered, and structured. (The structured type was not implemented.) Determining the type of attributes is one way the author contributes background knowledge that biases the learning process.

Decision rules and the description language

The rules learned by ILP can be viewed as having a production rule format:
⟨ConjunctiveClause⟩ ⟨CF⟩ ⇒ ⟨Prerequisite⟩
A certainty factor (CF) is associated with each clause. The certainty factor is an estimate of the probability that a student matching the clause really is in need of the prerequisite:

$$CF = \frac{POSCOV}{POSCOV + NEGCOV}$$

where POSCOV is the number of students matching (covered by) the clause who were found to lack mastery of the prerequisite, and NEGCOV is the number of students matching the clause who were found to have mastery of the prerequisite. Certainty factors are maintained by the system and are not entered or directly modified by the author.

In the recommendation phase, the student is compared to the left hand sides of all current rules. If none of the rules match, one prerequisite is chosen as a recommendation by a process to be described later. If one or more rules match, the prerequisite indicated by the matching rule with the greatest CF is automatically recommended. Other matching rules will result in a recommendation only if they have CFs greater than a threshold supplied by the course author.

There may be several rules relating to the same prerequisite (i.e. with the same right hand side). The disjunction of the left hand sides of these rules forms the description of students in need of that prerequisite. It is this disjunctive normal form (DNF) description that authors can initialize, observe, and modify. The description language is quite similar to the language used by Michalski and Chilausky (1980) and Lee and Ray (1986a). An example of a description is:

[Response='sugar'](6/7) OR
[Age=4..5 & Latency=3..8](2/4)

Nature of the search and guiding heuristics

Three goals or desired properties can be set for DNF descriptions learned from positive and negative examples:
° Consistency, which is satisfied by the exclusion of all negative examples.
° Completeness, which is satisfied by the inclusion of all positive examples.
° Parsimony, which is satisfied by minimizing the number of clauses.

From the author's viewpoint, ILP maintains a current best description (CBD) for each prerequisite. CBD is obtained by a beam search (Hayes-Roth & McDermott, 1976; Michalski, 1983) which is invoked immediately after a new example is acquired. The beam search starts with the previous CBD and applies generalization or specialization operations to discover new descriptions, one of which may be installed as the new CBD if it is better than the previous CBD. The author can initialize the CBD, and modify it at any time. If the author does not supply an initial description, the CBD is initialized by the system to be a description with zero clauses.

The author has control over the search through four parameters which can be changed at any time:
° maximum beam width (max_width)
° maximum depth of search (max_depth)
° maximum number of clauses (max_clause)
° a weight used in the preference criterion (w)

In order to guide learning and calculate certainty factors, ILP maintains four lists of examples for each description: PIN (containing pin_num positive examples included by the description), PEX (containing pex_num positive examples excluded by the description), NIN (containing nin_num negative examples included by the description), and NEX (containing nex_num negative examples excluded by the description). Each member of PEX violates the completeness of the description, and each member of NIN violates its consistency. Thus, descriptions are ranked for pruning during the search by a simple cost:

$$pex_num + (w * nin_num)$$

Among descriptions with the same cost, those with fewer clauses are preferred.

Generalization operations are passed a description and a positive example excluded by the description, and create new descriptions which include the example. Specialization operations are passed a description and a negative example included by the description, and create new descriptions which exclude the example. The operations roughly correspond to Michalski's (1983) generalization and specialization rules. The decision of what type of operations to apply (i.e. generalization or specialization) is determined by a comparison of pex_num and nin_num.

All descriptions created in the search have exactly the same set of examples as the CBD, although the set may be partitioned between the four example lists differently. For each description, examples in the PEX and NIN lists have a flag indicating whether they have been tried. When an example is used for generalization or specialization it is set to TRIED. New examples are set to NOT_TRIED before being pushed on to the front of the appropriate example list for the CBD. When a new description is created all its examples are initialized to NOT_TRIED. An example will only be used for generalization or specialization if it is NOT_TRIED.

In an early version of ILP a new example not matched by any CBD was subject to a random recommendation. But it often happens that such an example is 'closer' to one CBD, and turns out to be a positive example resulting in a generalization of that CBD. A heuristic, called pre-generalization, was developed which recommends the prerequisite whose CBD is most easily generalized to cover the non-matching example. When a clause is generalized it covers one or more positive examples that were not previously covered by any clauses in the same CBD. We denote the number of these examples as POS_GAINED. The same generalization may also force the clause to cover negative examples which the clause did not previously cover. We denote the number of such negative examples as NEG_GAINED. In pre-generalization, all clauses of all CBDs are temporarily generalized to cover the non-matching example. The prerequisite whose CBD has the pre-generalized clause scoring highest on the following criterion is recommended:

$$\frac{POS_GAINED}{POS_GAINED + NEG_GAINED}$$

Experiments with ILP

Four sets of examples, called cases, were constructed to test the performance of ILP. The cases were attempts to simulate instructional situations for the purpose of: (1) catching any unanticipated behavior of ILP, (2) providing an empirical basis for improvements to ILP, (3)determining practical parameter settings, and (4) suggesting ways that the author can most successfully interact with ILP. The basic strategy for creating the cases was to invent plausible conditions, stated as descriptions in the description language, under which students would fail an imagined objective. These descriptions were then used to select, from a large randomly generated set of simulated student models (i.e. attribute vectors), those student models failing the objective. ILP was presented with the problem of recovering correct descriptions given only the student models selected. The primary measure of performance was the number of incorrect recommendations made over the entire set of selected examples.

The introduction of noise into the examples always has the effect of degrading the performance of a learning program. In programs which assume all examples to be valid, the effect of noise can be catastrophic because a single invalid example may contradict and defeat a description which has been built upon many valid examples.

In order to investigate how ILP fares in the presence of noise, the cases were corrupted with a given n% of noise. This means that attribute values comprising examples in the cases had a n/100 probability of being altered. Figure 1 shows how the performance of ILP for one case deteriorated with increasing levels of noise. The y axis is the number of accumulated errors (i.e. incorrect recommendations). Learning is evidenced by the gradual flattening of the bottom three series. The top series, showing what happens when the learning procedure is disengaged, is included for comparison.

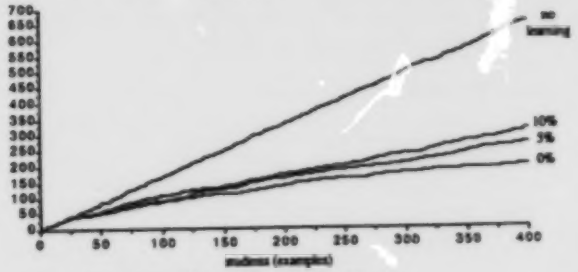


Figure 1. Performance with 0%, 5%, and 10% noise.

Conclusion

The following guidelines for interacting with ILP were suggested by the experimental evaluation:
° Nominal attributes with large value sets will swamp the search process and should be avoided.
° The attributes should be restricted to the few judged by the author to be most relevant.
° Descriptions entered by the author before any examples are available are quite unstable. For this reason the initial stages of the learning process should be monitored closely, to enable the author to re-enter descriptions when deemed necessary.

The instability of initial descriptions in the presence of noisy examples was the major shortcoming of ILP revealed by the experiments. One possible solution is to combine description initialization with Lee and Ray's (1986b)example set initialization method. Artificial examples generated from the description would lend stability until enough real examples are acquired, and thus prevent a description from being struck down prematurely by the first few invalid examples. To avoid the perpetuation of an incorrect initial hypothesis, the artificial examples could be gradually deleted as real examples are encountered.

Another shortcoming of ILP is that the clauses it produces tend to contain more terms than necessary. The bias toward comprehensibility (Michalski, 1983) should encompass a preference, not only for the least number of clauses, but also for clauses with the least number of terms.

References

Bundy, A., Silver, B. & Plummer, D. (1985). An analytical comparison of some rule-learning programs. Artificial Intelligence, 27, 137-181.

Hartley, J. (1973). The design and evaluation of an adaptive teaching system. International Journal of Man-Machine Studies, 5, 421-436.

Heines, J. & O'Shea, T. (1985). The design of a rule-based CAI tutorial. International Journal of Man-Machine Studies, 23, 1-25.

Hayes-Roth, F. & McDermott, J. (1976). Learning structured patterns from examples. 3rd international joint conference on pattern recognition, 419-423.

Kimball, R. (1982). A self-improving tutor for sym... In D. Sleeman & J.S. Brown (Eds.), Intelligent tutoring systems. Academic Press.

Lee, W. & Ray, S. (1986a). Probabilistic rule generator: A new methodology of variable-valued logic synthesis. Report UIUCDCS-R-86-1264. Dept. of Computer Science, University of Illinois.

Lee, W. & Ray, S. (1986b). Rule refinement using the probabilistic rule generator. AAAI-86, 442-447.

Michalski, R. & Chilausky, R. (1980). Learning by being told and learning from examples. International Journal of Policy Analysis and Information Systems, 4(2), 125-161.

Michalski, R. (1983). A theory and methodology of inductive learning. Artificial Intelligence, 20, 111-161.

Mitchell, T., Utgoff, P. & Banerji, R. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. In R. Michalski et al. (Eds.), Machine Learning (V. 1). Kaufmann.

Nesbit, J. (1988). Applications of learning hierarchies in adaptive instructional systems. Doctoral dissertation, University of Alberta.

Nesbit, J. & Hunka, S. (1988). An inductive learning procedure for adaptive instructional systems. RIR-88-5. Division of Educational Research Services, University of Alberta.

O'Shea, T. (1982). A self-improving quadratic tutor. In D. Sleeman & J.S. Brown (Eds.), Intelligent tutoring systems. Academic Press.

Quinlan, J.R. (1983). Learning efficient classification procedures and their application to chess end games. In R. Michalski et al. (Eds.), Machine Learning (V. 1). Kaufmann.

DEVELOPING A LEARNER MODEL FOR INTELLIGENT CAL: A SYSTEMS APPROACH

P.D. Mitchell
Centre for System Research and Knowledge Engineering
Concordia University
Montreal, PQ

Relatively little research exists pertaining to models of learners, hence few attempts have been made to integrate a sophisticated model into an Intelligent Tutoring System (ITS). Our research models the learner as an adaptive, self-organizing system in interplay with an instructional system. Like an ITS, the learner is represented as: a knowledge base; an inference system (for problem solving, goal setting, conceptual strategies); and a tutorial model (for interacting with other systems).

Individual learners behave differently when confronted with a learning task. Some approach tasks as if to understand and explain it; others, as if to acquire and remember information without much understanding. An ITS needs a model of learning which represents and adapts differentially both to prior knowledge and to different conceptual styles. Our model also includes the possibility of monitoring and shifting conceptual strategies.

Research has revealed that a learner's conceptual style predicts the level of understanding. Our cybernetic model of the learner would allow the instructional designer to compose different kinds of learning tasks, by posing different questions or problems to suit the learner's style, with the intent of maximizing understanding. The model of learning described is being incorporated into an ACME based tutoring system which will be evaluated.

DÉVELOPPEMENT D'UN MODÈLE D'ÉTUDIANT POUR EAO INTELLIGENT : ÉTUDE GLOBALE

On dénombre relativement peu de recherches axées sur la définition de modèles de l'étudiant; d'où, le petit nombre d'essais d'intégration d'un modèle à un tutoriel intelligent (TI). Notre recherche modélise l'étudiant comme un système adaptatif et automodifiant en interaction avec un système d'apprentissage. Tout comme le TI, l'étudiant a comme éléments : une base de connaissances, une machine à inférence (pour la résolution des problèmes, la détermination des buts, les stratégies conceptuelles) et un modèle de tutoriel (pour l'interaction avec d'autres systèmes).

Les étudiants individuels réagissent différemment en fonction de la tâche d'apprentissage à laquelle on les soumet. Certains abordent les tâches comme si on leur demandait de les comprendre et de les expliquer; d'autres, comme si on leur demandait d'acquérir et de conserver des connaissances sans aucune compréhension. Un TI nécessite un modèle d'apprentissage afin d'établir des représentations et de s'adapter aussi bien aux connaissances préalables qu'aux différents styles conceptuels. Notre modèle permet aussi le contrôle et la dérive des stratégies conceptuelles.

Notre recherche a démontré qu'un style conceptuel d'étudiant permettait de prédire le niveau de compréhension. Notre modèle cybernétique de l'étudiant est un outil qui permet au concepteur de cours de maximiser la compréhension en composant divers types de tâches d'apprentissage, c'est-à-dire poser des questions différentes ou proposer des problèmes adaptés au style de l'étudiant. Le modèle d'apprentissage décrit dans cette communication se trouve dans un tutoriel basé sur ACME que l'on se propose d'évaluer.

DEVELOPING A LEARNER MODEL FOR INTELLIGENT CAL: A SYSTEMS APPROACH

P. David Mitchell

Centre for System Research and Knowledge Engineering
Graduate Programme in Educational Technology
Concordia University, Montreal H3G 1M8

INTRODUCTION

Descriptions of Intelligent Tutoring Systems typically include not only a knowledge representation and a tutorial model but also assert the need for a model of the learner. Yet relatively little research exists pertaining to methodological issues in developing models of learners.

Our research models a learner as an adaptive, self-organizaing system in interplay with a tutorial system. The latter, in an ITS, includes a knowledge base and an inference system for using this knowledge and a tutorial system for interacting with the learner. The learner, too, may be modelled in a similar fashion as: a knowledge base (prior knowledge); an inference system (for problem solving, goal setting, learning strategies), and a tutorial model (for interacting with other systems).

Different learners may behave differently when confronted with a learning task. Some act as if interested in comprehending something in order to understand and explain it. Others seem to acquire and remember information without much understanding. Therefore we need a model of learning which represents and adapts differentially to prior knowledge as well as to different conceptual styles (including the possibility of monitoring and shifting conceptual strategies). Moreover, if the learner is to navigate through a knowledge base, the model must include representations of goal space, outcome space, beliefs, options and potential. This model also includes the possibility of monitoring and potentially shifting aims and conceptual strategies.

Given this model, instructional designers might compose different kinds of learning tasks or pose different questions or problems to the learner in order to maximize learning rate. This could be especially frutiful in a hypermedia environment.

The model of learning described is being incorporated into the design of an intelligent tutoring system which builds upon an earlier hypermedia system, C/CASTE (Mitchell, 1988; Mitchell & Dalkir, 1986).

OBSERVING IN AN INFORMATION SPACE

Blinded by the shimmering vision of AV display systems (the computer and interactive video being the latest) and a misguided enthusiasm for improving instruction, we too frequently overlook the role of the learner. We need to re-assess educational technology from the perspective of the learner and within the framework of cybernetic control theory (cf. Mitchell, 1989).

One the other side of the coin from imparting knowledge is learning or observing. And of course tutorial conversation requires frequent role reversals. But do people learn in essentially the same way?

That there are different conceptual styles has been realized for millenia and oft-identified as consisting of two major types, rote memory and understanding. As long ago as in classical Greece, people developed memory systems that also catered to visual or auditory conceptual styles, the former relying on vivid mental imagery (Yates, 1966). Many mnemonic systems have been devised to help people acquire and reproduce knowledge.

Each involves inner representations of what is learned but the representations may differ radically. Unfortunately these seem not to have permeated institutions dedicated to education and training. Such organizations tend to emphasize not how to learn and to think but what to learn and to think.

By way of illustration, consider Perry's (1959) finding that virtually all of his large sample of Harvard students were able to read a chapter in a book very rapidly. They mastered an impressive amount of detail as measured

by multiple-choice questions. Yet only one per cent could write a short statement indicating what the chapter was about. Such 'obedient purposelessness' underscores the need for learners to orient themselves toward an appropriate purpose when studying and not simply implement a strategy that is effective for another purpose.

Perhaps what is salient here is the learners' expectations. After all, the learner is essentially an observer and, as such, sets constraints on what he or she intends to observe and what they cannot observe. How exactly do they perceive the situation? Without speculating on this particular investigation, it is apparent that people have different conceptions of what is meant by learning. These may include memorizing, acquiring information, acquiring useful information, problem solving, discovery, thinking or creativity. In essence, people seem likely to think of learning either as something that happens to you (e.g. as you read, repeat, recite) or as something that you do actively (e.g. searching for facts, associating new ideas with existing knowledge, questioning the author). (The work of Ference Marton and his group has been particularly productive in exploring this problem; Surely complex human learning, to be significant, is less a process of (more-or-less) passive receiving of context-free facts than an active, goal-directed though context-determined search for facts -- indeed for anticipated facts.

Is there a relation between the perspective and approach a student takes to a learning task and the outcome of learning? Anecdotal reports about course-focused vs. interest-focused students abound; the former is caricatured as asking if "this" will be tested on the exam whereas the latter reads widely and frequently discusses the subject with others (cf. hobbiests vs. persons required to master the same subject). Such a qualitatively different approach seems likely to produce qualitative differences in understanding and comprehending between those for whom learning is intrinsically worthwhile and those for whom it is simply a job to be accomplished.

OBSOLETE INPUT-OUTPUT MODELS

Two key conceptual roots of educational technology are behavioural psychology and audiovisual media. The AV movement was concerned primarily with the organization and presentation of information to the learner. Interactive video and CAL offer potential but too frequently serve mainly as information display systems. Inheriting this stance, in instructional design too there is a profound concern with packaging and imparting information.

Skinner's experimental analysis of behaviour showed that the learner is more than a receiver of information and must engage in an activity (especially important in learning to study, to think, to be creative, to develop a love of the subject). In operant conditioning information may be presented but the key input is the consequence of the learner's behaviour. This contingent consequence may be mediated by the behavioural engineer as a function of his comparing the observed behaviour with intended behaviour. Equally it may be accidental or arranged by the learner.

Educational technology has inherited and combined these approaches. A stimulus-response, input-output model seems to prevail, where much effort goes into information display and the learner's output is compared with the system's behavioural objectives. In effect, educational technology has attempted to design systems that turn the learner into a programmable machine.

SELF-REGULATION

Skinner's model is essentially a cybernetic model insofar as it ignores environmental input and stresses the learner's behaviour and its effect on the environment. That effect -- called feedback in cybernetics -- may increase or decrease the probability of further behaviour of the same class. It is important to note that the consequence serves to strengthen or weaken the antecedant behaviour as a result of the individual's perception of it, not necessarily as an attribute of the consequent event. To illustrate, hungry rats will press a lever to get food; after eating they do not. The reinforcing quality of the food is contextual.

A closed chain of causal relationships might follow: the learner's behaviour "produces" an environmental change which in turn "produces" similar behaviour and so on, in a cycle that feeds on itself until a limiting resource is used up. A profound insight reveals the most powerful aspect of feedback: the organism actually "causes" its own behaviour. (Sometimes it does so in an environment designed to promote such learning.) Moreover, the absence of universal reinforcers in educational settings underscores the observation (of such investigators as Kelly, Rogers or Snygg and Combs) that behaviour is a function of the individual's personal frame of reference, his perception of himself and his environment and the meanings he attributes to them. In cybernetic terms, the person's behaviour controls his perception in relation to his intentions.

Thus we shift from an input-output model to a cybernetic model dominated by feedback which the observing system uses to control its own behaviour and thereby to attempt to control the behaviour of another system. Note that feedback monitors goal-directed behaviour; i.e. the system begins with some desired state or goal which is compared with its perception of the actual state during or following a behaviour episode. In effect, environment is what the receptors and brain perceive, i.e. an internal representation. Not only objects and events but also symbols and relations may be represented by these internal models. Internal events probably are represented in the same way and we assume that behaviour (overt or covert) acts on the inner as well as the outer environment. Powers (1973) demonstrated that what we control is our own input; our behaviour is the means of control. Expressed otherwise, the purpose of our action is to control our internal model of the perceived world. This is a powerful insight for educational technology to exploit. It opens the door not only to design of intelligent CAL but also to the design of new organizational structures for education.

In a tutorial conversation, two cybernetic systems become coupled (until a resource, e.g. time or attention, is used up) to form a new interacting system in which each begins with goals that it attempts to satisfy by monitoring the effects of its own behaviour on the other. Recall the cartoon showing two rats in a Skinner box. One says, "Boy, have I got this guy conditioned. Every time I press this lever he gives me food." This cartoon superbly introduces the idea of reciprocal control and cybernetics.

MODELS OF GOALS AND PERFORMANCE

Thus, instead of the input-output model, we conclude that behaviour is not so much a function of the environmental input as of a self-conscious "I" of each person in interplay not with his environment per se but with his perceptual model of that environment. The would-be regulator of all this, a human or an intelligent CAL system, mirrors the same process; the instructional system must have minimally a model of the subject matter, a model of the student's knowledge and conceptual style, and a model of communications and control strategies to respond to the student's behaviour (Mitchell, 1982).

It is noteworthy that research on learning styles shows that a decisive factor is the learner's approach. The best students: feel themselves to be the agents; construct and re-construct their private representations of knowledge, linking new knowledge to prior knowledge; and motivate and monitor themselves. This model of Oneself also is consistent with the Gothenburg research which shows that teaching, research on learning, and learning itself are all dependent on the student's conception of the subject matter and the model of learning held (i.e. one's model determines how we characterize the structure of experience and therefore the depth of understanding). Linked to knowledge engineering, this new model or paradigm could form the core of an expert system en route to artificially intelligent CAL.

SELF-CONTROL AND SELF-INSTRUCTION

If the purpose of one's action is to control the perceived world, a cybernetic model of oneself as learner, based on Stafford Beer's pioneering work, may prove useful (Beer, 1982;1983;1984). This model is consistent with research in psychology and education but begins with a different perspective.

At its heart is a perceptual field or set of relationships which determines that this is "oneself". Think of a human being not as mind, body, spirit or social unit but as "an entire and

interactive system". Oneself is an exceedingly complex, probabilistic system that maintains stability and integrity by virtue of an organizing principle, a set of relationships which determine that this is Oneself, not another self. Beer examines the instrinsic regulatory mechanism that holds everything together, maintaining one's identity.

Goals and their Achievement

The self-regulatory capacity of the body seems automatic but what if one aspires to be different (e.g. run a marathon, read 5,000 words per minute)? As Beer shows, such pursuits require extending the self-regulatory capacity of both body and mind, i.e. of the cybercyte. Thus I may have the potential to run a marathon or to read at 5,000 wpm but I lack the regulatory model required. If Oneself sets goals and aspires to achieve them, then Oneself must change one's model of oneself. Why? Because things one is only potentially capable of doing are not initially included in one's regulatory model. There is a spectrum of options from which to choose, e.g.: actions, models, beliefs and aspirations.

Recall that the purpose of action is to control the perceived world by comparing this model with an internal model of a desired end-state. It is essential therefore to establish a goal-state, to believe it can be achieved and will be achieved, and to visualize oneself already in the goal state -- and then to act accordingly. At this point the regulatory system should respond to perceived deviations from that goal.

Beer's concept of selfhood thus advocates self-improvement -- and, by extension, education and societal improvement -- based on the existence of autonomous regulatory mechanisms that permit self-control. However the rules which govern the effectiveness of this self-contrel require the regulatory process to generate new states and detect and store patterns that can reduce discrepancies.

Research has revealed that a learner's level of understanding is affected by his/her approach to learning. We assume that the goal of the instructional designer is to create learning experiences that will result in an efficient and thorough understanding of a topic. A cybernetic model of the learner and of the ITS would allow the instructional designer to compose different kinds of learning tasks and opportunities, by posing different questions or problems to suit the learner's style, with the intent of monitoring and responding to misconceptions, thus enhancing understanding (Mitchell, 1988).

Such knowledge engineering may be extremely valuable to designers of learning opportunities, especially in the context of CAL Interactive Video systems. For learning design, we need to develop knowledge based systems such that a learner may navigate through knowledge, choosing an idiosyncratic route to his chosen goal. Our computer based Abstract Concept Modelling and Evaluation system (Mitchell, 1987) and Concordia Course Assembly System and Tutorial Environment (Mitchell & Dalkir, 1986) provide a working prototype for individualized learning.

REFERENCES

Beer, S. (1982) Man in a garrulous silence. Unpublished ms.

Beer, S. (1983) The will of the people. Journal of the Operational Research Society, 34, (8), 797-810.

Beer, S. (1984) The cybernetics of selfhood. Plenary address to Seventh European Meeting on Cybernetics and Systems Research, Vienna.

Marton, F. & Saljo, R. (1976) On qualitative differences in learning I: Outcome and process. British Journal of Educational Psychology, 46, 115-127.

Mitchell, P.D. (1975) The discernible educational technolgist. Programmed Learning and Educational Technology, 12, (5), 306-325.

Mitchell, P.D. (1982) Representation of knowledge in CAL courseware, Comput.& Educ., 6, 61-66

Mitchell, P.D. (1987) ACME: An aid to tutorial development and evaluation of understanding. Presented to CAL 87 Symposium on Computer Assisted Learning, Glasgow (April).

Mitchell, P.D. & K. Dalkir (1986) C/CASTE: An artificial intelligence based computer aided learning system, in Computer Assisted Learning, the next five years. Ottawa: National Research Council

Mitchell, P.D. (1989) The Future of Educational Technology is Past, in Canadian Journal of Education and Communication, Vo. 18, No.1, 3-27.

Perry, W. G. (1959) Students use and misuse of reading skills, Harvard Educational Review, Vol. 29, No. 3, 193-200.

Powers, W.T. (1973) Behavior: The Control of Perception. Aldine, New York.

Yates, F.A. (1966) The Art of Memory. London: Routledge & Kegan Paul.

MODÉLISATION DE L'ÉTUDIANT: UNE APPROCHE CONTEXTUELLE

C. Frasson
Dép. d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, PQ

B. de La Passardière
Laboratoire MASI
Université Pierre et Marie Curie
Paris, France

Dans le présent article, nous présentons un modèle de l'étudiant destiné à être intégré dans un système d'Enseignement Intelligemment Assisté par Ordinateur (EIAO). L'approche utilisée ne se limite pas à la prise en considération des connaissances incomprises ou manquantes, mais intègre le contexte dans lequel ces connaissances ont été acquises. Pour ce faire, nous élaborons des modèles hypothétiques et proposons une méthode qui, par comparaison de ces derniers et de situations réelles, permet une modélisation plus précise de l'étudiant et une évaluation du degré de fiabilité de cette modélisation. Des méthodes d'apprentissage plus efficaces peuvent alors être sélectionnées.

Nous montrons comment un tel modèle peut être implémenté en ART.

STUDENT MODELING: A CONTEXTUAL APPROACH

In this paper, we present a student model which can be integrated in an Intelligent Tutoring System (ITS). The approach is not restricted to misconceptions or missing conceptions but consider the context in which the knowledge was obtained. Hypothetical models are built. We propose a method based on comparison of these models and real situations. We then obtain a more precise model, so as more efficient learning methods can be chosen.

Furthermore, we show how such a model can be implemented in ART.

MODELISATION DE L'ETUDIANT : une approche contextuelle

C. Frasson*, B. de La Passardière**

* Université de Montréal
Dépt IRO C.P. 6128
Montréal H3C 3J7

** Université Pierre et Marie Curie
4, place Jussieu
75230 Paris Cedex 05

1. INTRODUCTION

Les systèmes tuteurs intelligents (STI) visent à incorporer au sein d'un même système, l'expertise de plusieurs spécialistes afin de fournir à l'étudiant un enseignement adapté à son profil. Globalement, l'élaboration de tels systèmes nécessite de rassembler auprès d'un informaticien, un spécialiste de la matière enseignée, un spécialiste en pédagogie et enfin un spécialiste en connaissances humaines. En effet, il est très rare qu'un seul humain réunisse effectivement les trois expertises. Par contre, le domaine d'enseignement couvert par ce dernier est généralement plus large que celui abordé par un tuteur intelligent.

La tâche du spécialiste en connaissances humaines consiste à déterminer les caractéristiques de l'étudiant et à préciser les conditions d'acquisition de ses connaissances. Dans un STI, cette tâche est assurée par le **modèle étudiant** qui établit un diagnostic sur l'étudiant au fur et à mesure des interactions qu'il a avec le système. Une différence notable entre un enseignant et un STI est que le premier agit, généralement, au niveau d'un groupe alors que le second est susceptible d'agir au niveau de chaque étudiant. Par conséquent, il est important que le STI dispose d'une perception plus fine des caractéristiques de l'étudiant tant du point de vue psychologique que du point de vue de ses connaissances.

Le modèle étudiant a un rôle important dans le fonctionnement d'un STI dans la mesure où les actions entreprises par ce dernier dépendent fortement de la connaissance de l'état d'avancement de l'étudiant dans son processus d'apprentissage. Ce processus repose souvent sur un curriculum qui comporte tous les sujets disponibles de la matière, ainsi que leurs difficultés respectives et les préalables nécessaires. Le passage d'un sujet à un autre est réalisé en interrogeant périodiquement le modèle étudiant. En effet, le STI décide du prochain sujet à présenter en fonction des résultats observés. Certaines approches pour réaliser des STI [Imb 88] utilisent une structure de curriculum qui inclut une partie du modèle étudiant comprenant ses résultats et ses difficultés. Cette solution permet de naviguer plus aisément dans le curriculum qui s'adapte graduellement à l'étudiant au cours de la session de travail.

Dans un STI, le modèle étudiant peut être utilisé à différentes fins. Van Lehn [Pol 88] distingue plusieurs utilisations possibles : pour évaluer la maîtrise de l'étudiant sur un sujet et déterminer la suite en conséquence ; pour choisir le moment le plus adéquat pour intervenir , pour générer des exercices d'un niveau approprié ; pour fournir des explications compréhensibles ; ou encore pour adopter les meilleurs choix pédagogiques. De plus, Van Lehn propose une classification du modèle étudiant selon trois critères : la plage des informations disponibles (les états mentaux, les états intermédiaires, les états finals), le type de connaissances traitées (les connaissances déclaratives, les connaissances procédurales linéaires, les connaissances procédurales hiérarchisés), et enfin la nature du modèle, c'est à dire la méthode par laquelle le modèle prend en compte l'écart entre l'Expert et l'Etudiant (recouvrement, correction basée sur un catalogue d'erreurs, réparation par reconstitution des erreurs à partir d'erreurs partielles). Toutes ces approches accordent une certaine fiabilité au modèle étudiant et conduisent l'enseignement en se fiant à celui-ci. Or, des déformations considérables quant aux actions à entreprendre peuvent se produire si le modèle n'est plus adéquat.

L'approche que nous proposons ne peut s'inclure dans aucune des catégories précédentes. L'utilisation que nous faisons du modèle étudiant est certes de permettre au module Tuteur du STI de déterminer la suite des interventions pédagogiques, mais surtout d'optimiser ce module, c'est à dire

d'en accélérer le processus de décision. De plus, nous pensons que l'adéquation du modèle doit être continuellement vérifiée et remise en cause. C'est pourquoi, nous ne limitons pas le modèle aux seules connaissances acquises, incomprises ou incomplètes. En effet, nous intégrons dans ce modèle le **contexte** dans lequel ces connaissances ont été acquises. A chaque étape du processus didactique, nous émettons un ensemble d'hypothèses non seulement sur le savoir de l'étudiant, mais aussi sur la nature de l'apprentissage des connaissances, et évaluons la **pertinence** du modèle étudiant.

Après avoir donné les caractéristiques du modèle et précisé l'importance du contexte, nous indiquons comment il peut être implémenté en utilisant le logiciel d'Intelligence Artificielle ART [Inf 86]. Nous montrons ensuite le fonctionnement général du modèle.

2. MODELE CONTEXTUEL

L'acquisition de connaissances est variable selon les individus. Un des moyens pour en apprécier le degré est de confronter l'étudiant à des situations plus ou moins complexes portant sur la connaissance enseignée.

Lorsqu'on délivre un enseignement, il est possible,de déterminer pour chaque activité le contexte dans lequel s'est déroulée cette activité et les résultats obtenus. Il est, par contre, plus difficile de déterminer le contexte d'une connaissance qui a été acquise par l'étudiant auparavant, dans un contexte éducatif inconnu ou peu connu. Seules des hypothèses sont alors permises.

Le modèle étudiant est basé sur des schémas de type :

$$S = (K, C, E)$$

où K est une connaissance, C un contexte, et E une évaluation.

- La *connaissance* K constitue l'élément de base du curriculum. Elle correspond soit à un savoir, soit à un savoir-faire ou encore à un savoir-utiliser.
- Le *contexte* C, dont il est ici question, correspond au contexte d'apprentissage. Il reflète l'état d'avancement de l'étudiant dans le processus d'apprentissage de la connaissance, selon la classification de Gagné [Gag 76, Gag 84]. Dans cette classification, on trouve plusieurs niveaux, qui vont de la motivation de l'étudiant vis à vis de la connaissance, à l'encodage, la mémorisation, le transfert et le renforcement.
- L'*évaluation* E peut être soit l'évaluation hypothétique E_H, déduite par le système en fonction du modèle étudiant, soit l'évaluation réelle E_R, résultant de l'activité que l'étudiant vient de réaliser. L'évaluation hypothétique correspond à la perception qu'à le système de l'étudiant après une série d'activités, et concerne des connaissances préalables ou des connaissances qui n'ont pas encore été évaluées effectivement.

On distingue donc dans le modèle étudiant, deux éléments fondamentaux. D'une part, le *schéma réel* :

$$S_R = (K, C, E_R)$$

qui rend compte de l'évaluation de la connaissance K dans le contexte C, et d'autre part, le *schéma hypothétique* :

$$S_H = (K, C, E_H)$$

qui, lui, établit la croyance (au niveau du savoir de l'étudiant) qu'à le système sur la connaissance K dans le contexte C.
L'ensemble des schémas de type S_R constitue l'*historique* de l'activité de l'étudiant, tandis que l'ensemble des schémas de type S_H reflète à un instant donné, les différentes *hypothèses* émises sur les connaissances de l'étudiant.

Pour établir et mettre à jour les hypothèses, le curriculum intègre des informations qui indiquent le *contexte de cohérence*, c'est à dire le degré de liaison unissant les différentes connaissances, et le *contexte d'apprentissage* dans lequel la connaissance est supposée être connue. Au début de la session de travail, le curriculum est initialisé en fonction du niveau d'apprentissage requis (expert, intermédiaire, novice...), et donc des objectifs à atteindre. Cette initialisation fixe les connaissances les unes par rapport aux autres, non seulement dans un niveau de cohérence, mais aussi dans un niveau d'apprentissage. Autrement dit, l'organisation du curriculum permet de déterminer ce qu'on appelle l'*environnement* de la connaissance, c'est à dire l'ensemble des connaissances et des contextes qui sont liées à une connaissance particulière :

$$EV = (C_a, C_c)$$

où C_c est le contexte de cohérence et C_a le contexte d'apprentissage (Figure 1).

Soit K_0 la connaissance, objet de la dernière activité de l'étudiant ;
On appelle *zone de connaissances* Z_0, un ensemble de connaissances, déterminé par le contexte de cohérence, liées à la connaissance K_0 :

$$Z_0 = \{K_1, K_2, ..., K_n\}$$

On appelle *hypothèse* H_0, l'ensemble des schémas hypothétiques qui traduisent les différentes hypothèses émises en référence à la zone Z_0 :

$$H_0 = \{(K_1, C_1, E_{H1}), (K_2, C_2, E_{H2}), ..., (K_n, C_n E_{Hn})\}$$

De plus, on appelle *pertinence* P, l'écart entre l'évaluation réelle et l'évaluation hypothétique d'une connaissance :

$$P = | E_R - E_H | \text{, avec } E_R \neq \emptyset$$

Sous ces conditions, on appelle *réseau d'hypothèses* R, la compilation à un instant donné de toutes les hypothèses émises. Autrement dit, le réseau d'hypothèses traduit l'évolution des hypothèses au cours du temps. En effet, celui-ci est périodiquement mis à jour en tenant compte de la pertinence des hypothèses, des contradictions entre les hypothèses, ou encore de la confirmation d'une hypothèse.
Ainsi, le modèle étudiant comprend en plus de la partie statique, comprenant les caractéristiques initiales de l'étudiant, une partie dynamique qui intègre l'historique de l'étudiant et le réseau d'hypothèses.

3. REPRÉSENTATION DU MODELE

Le modèle présenté fait fortement appel à des outils de gestion d'hypothèses. En conséquence, il faut soit utiliser des outils disponibles dans les systèmes dédiés à l'Intelligence Artificielle, soit développer des outils adaptés à ce type de problèmes. Pour notre part, nous utilisons comme outil de développement, le logiciel ART "Automated Reasoning Tool" [Inf 86] dont la structure de *point de vue* est un moyen puissant pour modéliser des hypothèses qui évoluent dans le temps. En outre, ART est un outil de développement de système expert, qui accepte la programmation à base de règles, qui intègre le chaînage avant et arrière, qui permet le raisonnement positif et négatif (faits présents et faits absents) et qui, de plus, est interfacé avec l'environnement Lisp. Autrement dit, ART offre une gamme de facilités qui s'adaptent bien à la représentation du réseau d'hypothèses tel que présenté et à son évolution. En effet, associé à cette notion de point de vue, il existe en ART un mécanisme d'héritage, ainsi que des règles de contrainte et de croyance pour mettre à jour la structure des points de vue. Ainsi, le réseau d'hypothèses n'est autre que l'arbre des points de vue, dont l'évolution est assurée par quatre grandes classes de régles :

• les **régles de création** de points de vue, qui sont du genre:

SI cet événement survient
ALORS créer un point de vue pour cette hypothèse

et qui, en ART, s'écrivent sous la forme :

```
(defrule création-hypothèse
   (liste des faits initiaux)
=>
   (hypothesize
     (assert (liste des suppositions))
     (assert (liste des faits))))
```

• les **régles de suppression** de points de vue, qui sont du genre :

SI cet événement survient
ALORS rejeter cette hypothèse

et qui, en ART, s'écrivent sous la forme :

```
(defrule suppression-hypothèse
  (declare (salience *constraint salience*)
  (liste des faits initiaux))
=>
  (poison "raison"))
```

• les **régles de confirmation** de points de vue, qui sont du genre :

SI cet événement survient
ALORS prendre cette hypothèse comme certaine

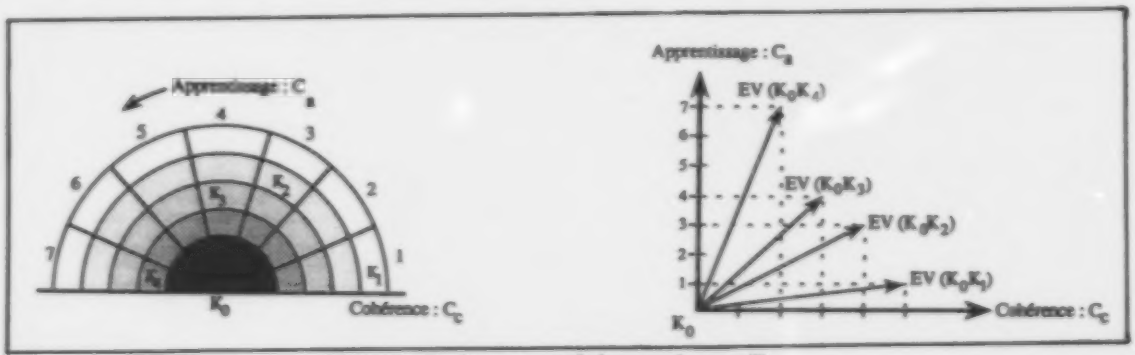


Figure 1 : Environnement de la connaissance K_0

et qui, en ART, s'écrivent sous la forme :

```
(defrule confirmation-hypothèse
   (liste des faits initiaux)
=>
   (reliaux ?root "raison")
```

• les **règles de fusion** de points de vue, qui sont du genre :

SI cet événement survient
et cet autre événement survient
ALORS fusionner les points de vue en une hypothèse

et qui, en ART, s'écrivent sous la forme :

```
(defrule fusion-hypothèse
   (viewpoint ?v1
      (liste des faits initiaux))
   (viewpoint ?v2
      (liste des faits initiaux))
=>
   (merge (?v1 ?v2)
      (assert (liste des faits))))
```

4. FONCTIONNEMENT GÉNÉRAL

Un des buts du modèle étudiant est de guider le module Tuteur du STI dans le choix de la stratégie pédagogique à adopter et des activités à proposer. Pour accomplir cette tâche, le Tuteur doit prendre en compte à la fois les objectifs à atteindre et le profil de l'étudiant. La validité de son choix dépend donc pour partie de la **fiabilité** du modèle étudiant. Or, l'approche contextuelle du modèle étudiant que nous proposons permet, avec son réseau d'hypothèses, de déterminer le degré de fiabilité du modèle. Ainsi, soit le modèle étudiant est considéré comme fiable, et le Tuteur se repose alors sur ses propres stratégies pédagogiques, soit il ne l'est pas, et dans ce cas, le Tuteur cherche à proposer des activités qui, avant tout, vont corriger et à améliorer le modèle. Il adopte cette stratégie jusqu'à ce qu'il puisse de nouveau se servir efficacement du modèle étudiant comme guide.

Le fonctionnement général du modèle étudiant comprend donc une phase de mise à jour du réseau d'hypothèses et une phase de détermination du degré de fiabilité. La première est réalisée par un ensemble de règles portant sur les schémas et les hypothèses, tandis que la seconde fait appel à des méta-règles portant sur la structure du réseau d'hypothèses. Ainsi, en se basant sur les grandes classes de règles énoncées précédemment, le réseau d'hypothèses est mis à jour en utilisant :

• des règles liées à l'évaluation effective d'une activité,qui prennent en compte la *pertinence* associée aux différentes connaissances mises en jeu par l'activité : règles de pertinence élevée, de pertinence moyenne, de pertinence faible ;

• des règles liées à l'*environnement* des connaissances évaluées, qui déterminent les hypothèses possibles : règles d'élaboration des hypothèses ;

• des règles liées à l'intégration des nouvelles hypothèses, qui déterminent pour chacun des schémas ceux qui sont renforcés, abandonnés ou modifiés : règles de conflits, de convergence, de divergence ;

• des méta-règles liées à la structure du réseau d'hypothèses, qui permettent d'évaluer l'évolution du réseau en termes de fiabilité : règles de fiabilité, de déséquilibre.

5. CONCLUSION

Le modèle étudiant proposé est basé sur une approche contextuelle dans laquelle on cherche à modéliser **ce que connaît** l'étudiant, mais aussi **dans quel contexte** il a acquis ces connaissances. Cette approche nous paraît importante dans la mesure où elle permet d'obtenir un modèle étudiant plus fiable et donc sur lequel, à certains moments du processus d'apprentissage, le Tuteur peut s'appuyer davantage. Cette adéquation entre le modèle et l'étudiant est rendue possible par l'implémentation du réseau d'hypothèses qui rend compte non seulement de **ce que l'on sait sur l'étudiant**, mais aussi de **ce que l'on croit savoir** qu'il sait et qui plus est à quel niveau il est susceptible de le savoir.

Ce type de modèle nous paraît offrir de nombreuses possibilités dans le cadre des tuteurs intelligents dans la mesure où il permet de mieux percevoir l'étudiant et donc de planifier de manière plus adéquate le type d'activités proposées. L'implémentation de ce modèle est prévue dans le prototype Word Tutor, un tuteur intelligent sur le traitement de texte développé dans le cadre du projet HERON à l'Université de Montréal.

RÉFÉRENCES

[Gag 76] Gagné R.M., *Les principes fondamentaux de l'apprentissage*, Les éditions HRW Ltee, Montréal (1978)

[Gag 84] Gagné R.M., *The conditions of learning*, 4 ed, Les éditions HRW Ltee, Montréal (1984)

[Imb 88] Imbeau G, Frasson C. & Gauthier G, *A Multi-Expert System for Large Scale Intelligent Tutoring*, Actes de la Conférence ITS-88, Montréal (Juin 1988)

[Inf 86] Inference Corporation, *ART Reference Manual*, (1986)

[Pol 88] Polson M., Richardson J.(Editors), *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Ass.(1988)

LOGIX: UN SYSTÈME DÉDUCTIF ET INDUCTIF À BASE DE CONNAISSANCES

G. Brulé, L. Thérien, G. Lévesque
Dép. de mathématiques et d'informatique
Université du Québec à Montréal
Montréal, PQ

Les humains raisonnent la plupart du temps de façon déductive alors que le mode inductif peut s'avérer une forme toute aussi riche pour l'acquisition de nouvelles connaissances. Par ailleurs les recherches en psychologie cognitive ont démontré qu'il fallait passer par un processus d'apprentissage explicite pour utiliser couramment le modus tollens.

La majorité des systèmes experts actuels se sert uniquement de la déduction et du modus ponens comme mode d'inférence. Nous avons construit un prototype de système à base de connaissances, fondé sur la logique des propositions, qui incorpore les modes déductif (modus ponens et modus tollens) et inductif (par recours à l'utilisateur) dans le but de créer un outil qui aide l'utilisateur à élargir ses possibilités de raisonnement.

LOGIX: A DEDUCTIVE AND INDUCTIVE KNOWLEDGE-BASED SYSTEM

Most of the time, humans think in a deductive way while the inductive mode could be an equally effective way of acquiring new knowledge. Research in cognitive psychology has shown that an in-depth training process is required in order to use the modus tollens fluently.

The majority of current expert systems use only deduction as an inference mechanism. We have developed a prototype of a knowledge-based system, based on propositional logic, which incorporates the deductive (modus ponens and modus tollens) and inductive (by user assistance) modes to create a tool that will assist the user to enlarge his reasoning capabilities.

LOGIX: un système déductif et inductif à base de connaissances

Gilles Brulé, Lyne Thérien et Ghislain Lévesque

Université du Québec à Montréal,
Montréal, Québec.

Introduction

Les travaux que nous effectuons dans le domaine des outils d'enseignement assisté par ordinateur s'inspirent d'objectifs comme la recherche de la simplicité et l'utilisation de moyens conventionnels; notre but est de rejoindre plus facilement l'étudiant là où il se trouve, tenant compte des moyens simples dont il dispose.

Ainsi le premier outil que nous avons développé, un tutoriel d'exercices concernant l'enseignement de la logique des propositions au niveau collégial[1] était fondé sur la logique des propositions et un processus de reconnaissance de formes dans des chaînes de caractères [Camarda, Lévesque, 87]; il s'inspirait des travaux déjà présentés par [Larvet, 84]. Ce tutoriel, écrit en mode procédural et fonctionnant avec un ordinateur personnel (IBM et compatible), comporte des exercices portant sur la qualité, la quantité et le type de propositions de forme canonique (sujet - verbe - prédicat), les opérations sur les propositions, la détermination de la figure et du mode du syllogisme, l'application valide des règles de conclusion et enfin la formulation et la validation de la conclusion du syllogisme. Ces travaux nous ont également amenés à développer les algorithmes de manipulation des huit formes de syllogismes hypothétiques ou conditionnels, soit les deux figures, le *modus ponens* et le *modus tollens*, et leur quatre modes respectifs [Chenique, 75 ; Suppes et Hill, 64].

Ces algorithmes constituaient alors le noyau du moteur d'inférence du squelette de système à base de connaissances que nous avons élaboré et que nous présentons sous le nom de LOGIX. Ce squelette fonctionne soit en chaînage avant, soit en vérification assistée [Brulé, Thérien et Tremblay, 88] et dépasse largement le modèle déjà présenté par [Larvet, 87]. Les règles utilisées par le système sont formulées comme suit :

si <antécédents> alors <conséquents>

où les antécédents peuvent prendre la forme d'une ou de plusieurs propositions reliées de manière conjonctives ou disjonctives, ou selon les deux formes à la fois, tandis que les conséquents sont reliés uniquement par des conjonctions. Les connaissances sont intégrées dans le système, sous la forme de règles et de faits, et sont représentées par des propositions canoniques , affirmatives ou négatives; elles sont manipulées par des procédures fondés sur la reconnaissance des formes.

La présentation qui va suivre décrit les modes de construction et de fonctionnement du moteur d'inférence et explique comment le prototype applique les modes de raisonnement déductif et inductif, tout en détectant les inconsistances possibles. Divers cas types seront présentés pour illustrer ce fonctionnement. Comme le moteur d'inférence implante le *modus tollens*, nous montrerons ensuite, sur la base des recherches récentes en psychologie cognitive, en quoi ce prototype peut s'avérer intéressant pour faciliter l'apprentissage du raisonnement par *modus tollens* et l'exercice du raisonnement inductif. Nous conclurons enfin en indiquant les limites que nous avons identifié à notre prototype et les développements futurs dont il fera l'objet.

[1] Correspond aux premières années d'études universitaires au Canada anglais.

Moteur d'inférence

Pour bien comprendre le fonctionnement du moteur d'inférence, il faut connaître la structure de la base de règles. La composition d'une règle ne fait pas appel à des variables ou quantificateurs, car le système est fondé sur la logique de propositions. Une règle est constituée d'un groupe d'antécédents et d'un groupe de conséquents. Ces groupes sont définis comme étant plusieurs propositions obtenues par la concaténation du sujet, du verbe et du prédicat. Le langage utilisé pour décrire les règles emploie le formalisme Backus-Naur[2] à la Figure 1.

```
BaseRègles ::= [ règle ]+
règle ::=  SI proposition [ [ ET | OU ] proposition ]*
           ALORS proposition [ proposition ]*
proposition ::= sujet [ NE | N' ]- verbe [ PAS ]- [ prédicat ]-
verbe ::= mot
sujet, prédicat ::= [ mot]+
```

Figure 1 : Grammaire de description des règles.

Voici quelques exemples de règles définies avec ce langage.

(1) **SI** le soleil brille
ET le ciel est bleu
ALORS il ne pleut pas

(2) **SI** les figures sont deux triangles rectangles isocèles
ET les figures partagent la même hypoténuse
OU la figure est un quadrilatère à quatre côtés égaux plus un angle droit
ALORS la figure est un carré

(3) **SI** les hommes n'arrêtent pas la pollution
ALORS l'espèce humaine ne survivera pas bien longtemps

[2] Notation.
+ : 1 ou plusieurs fois
* : 0 ou plusieurs fois
- : 0 ou 1 fois

Le système emploie une stratégie de type "irrévocable" [Farreny, 85] du fait que l'application d'une règle n'est jamais remise en cause. Par ailleurs, l'application d'une règle est guidée par la valeur de vérité caractérisant chacune de ses propositions, soit vraie, fausse et indéterminée. De plus, à partir du moment où un fait est confirmé ou infirmé, la valeur de vérité de ce dernier ne change plus, ce qui en fait un système de type "monotone" [Farreny, 85].

En parcourant les règles selon un mode séquentiel, le choix des règles à déclencher se fait par la reconnaissance de la première règle non-vérifiée de la base de règles. Considérant la valeur de vérité du groupe d'antécédents et de conséquents de cette règle, valeur prise à la suite de l'interprétation de la base de faits accessibles, le système choisira d'inférer à l'aide du *modus ponens* ou du *modus tollens*. Si la valeur déterminée correspond à l'affirmation du groupe d'antécédents, le *modus ponens* sera appliqué. Par contre, si cette évaluation démontre l'infirmation du groupe de conséquents, le *modus tollens* sera appliqué [Chenique, 75 ; Suppes et Hill, 64]. L'évaluation des deux groupes se fait en considérant les liens logiques reliant les propositions entre elles. Ces liens, conjonctifs ou disjonctifs, sont de priorité égale et traités séquentiellement. La formulation des règles, ne faisant pas appel à un système de parenthèses, nécessite que les prémisses reliées par un connecteur "ET" précèdent toujours celles qui sont reliées par des connecteurs "OU", s'il y a lieu.[3]

Si la valeur des faits composant la règle devait indiquer qu'il est possible d'utiliser les deux sortes d'inférence, une inconsistance serait détectée [Brulé, Thérien et Tremblay, 88]. Lorsque la règle est traitée selon le *modus ponens*, les conséquents de celle-ci deviennent des faits confirmés qui seront ajoutés à la base de faits. Si la règle doit être traitée selon le *modus tollens*, le moteur tentera d'infirmer les prémisses ayant une valeur indéterminée de façon à ce que tout le groupe d'antécédents soit infirmé, considérant les liens logiques existant entre les prémisses. Pour évaluer le groupe d'antécédents, tout en tentant de l'infirmer, le moteur se base sur la correspondance de la table logique décrite en Figure 2.

A	B	¬(A ∧ B)		¬A ∨ ¬B	¬A	¬B
V	V	F	⇔	F	F	F
V	F	V	⇔	V	F	V
F	V	V	⇔	V	V	F
F	F	V	⇔	V	V	V

Figure 2 : Table logique.

Ainsi l'expression (A ∧ B) → C présente dans la base de règles et le fait C infirmé dans la base de faits, permet d'appliquer le *modus tollens*. Le système tentera d'affirmer alors l'expression (¬A ∨ ¬B).

$$\frac{(A \wedge B) \rightarrow C \qquad \therefore \neg C}{(\neg A \vee \neg B)}$$

Pour ce faire, le système prendra en considération les valeurs de vérité associées aux faits A et B, pour ensuite inférer sur ces derniers s'ils sont indéterminés. L'infirmation du ou des faits contenus dans le groupe d'antécédents suit le cheminement dicté par le tableau logique de la Figure 3 qui est incorporé dans une structure arborescente.

Ainsi dépendamment de la valeur respective de ¬A et de ¬B, le système utilisera les branches nécessaires qui mèneront à un résultat final vrai. Lorsque toutes les possibilités sont

¬A	I	I	I	V	V	V	F	F	F
¬B	I	V	F	I	V	F	I	V	F
(¬A ∨ ¬B)	I	V	I	V	V	V	I	V	F

Figure 3 : Infirmation de (A ∧ B).

épuisées, s'il existe plusieurs façons d'affirmer (¬A ∨ ¬B), l'induction entre en jeu (Voir Induction par recours à l'utilisateur).

Dans tous les cas, s'il se produit une tentative de conclusion sur un fait ayant déjà une valeur contraire, une inconsistance sera relevée [Brulé, Thérien et Tremblay, 88].

A partir de la règle (A ∧ B ∨ C) → D, qui est la représentation symbolique de l'exemple (2), le moteur d'inférence appliquera le *modus ponens* advenant le cas où le fait C serait confirmé dans la base de faits; la confirmation du fait D représenterait le résultat obtenu. Pour la même règle, l'infirmation du fait D entraînerait l'application du *modus tollens*.

$$\frac{(A \wedge B \vee C) \rightarrow D \qquad \therefore \neg D}{(\neg A \vee \neg B \wedge \neg C)}$$

Après modification du groupe d'antécédents pour obtenir (¬A ∨ ¬B ∧ ¬C), les combinaisons suivantes sont possibles pour assurer la vérité de ce groupe d'antécédents:

a) ¬A et ¬C,
b) ¬B et ¬C,
c) ¬A et ¬B et ¬C.

L'usager devra alors apporter son opinion sur la valeur de vérité des faits indéterminés pour permettre de tirer les conclusions adéquates. Cette appel à la détermination des faits en présence ouvre à l'utilisateur la voie pour tester les alternatives possibles et connaître les résultats de leurs applications dans la suite du raisonnement. Cette forme d'induction est rendue possible lors de l'application du *modus tollens* par notre système. (Voir Induction par recours à l'utilisateur)

Structure des données

La structure interne des données qui permet de traiter les règles et faits de cette façon est présentée à la figure 4 et la description de chacune des structures suit:

(a) Arbre de règles : arbre binaire balancé représentant chacune des règles composant la base de règles. Ses éléments sont composés d'un numéro de règle, d'un indicateur de vérification et d'un pointeur à la première prémisse. Les règles sont ordonnées alphabétiquement sur la base de la première prémisse (i.e. concaténation des sujet, verbe et prédicat) d'une règle donnée.

(b) Arbre index : Cet arbre est formé lors de la lecture des règles et contient la première occurence de chacun des faits que le système rencontre dans les règles. Ainsi, la valeur de vérité de chacun des faits est évaluée en fonction de cet index en tenant compte du facteur de négation appliqué sur le verbe. Dans notre exemple, le fait "Il ne pleut pas" sera affirmé, alors que

[3] Les parenthèses ne sont pas implantées dans la version actuelle du prototype (Voir Développements futurs).

son correspondant dans l'arbre d'index pourrait être "Il pleut" qui prendrait alors la valeur infirmée. L'arbre d'index contient donc la base de faits représentant toutes les propositions constituant les règles.

(c) Liste de propositions formant une règle : Chacun des éléments de cette liste bidirectionnelle contient l'état de chaque proposition: prémisse ou conclusion (i.e. P ou C), affirmation ou négation du verbe, la valeur de vérité associée (i.e. Confirmée, Infirmée ou Indéterminée), le lien logique avec la proposition précédente, ainsi que son contenu syntaxique (i.e. concaténation des sujet, verbe et prédicat).

(d) (e) (f) Arbre de sujets, de verbes, de prédicats : Ces arbres sont produits lors de la lecture de la base de règles. Il renferme tous les sujets, verbes et prédicats contenus dans les propositions.

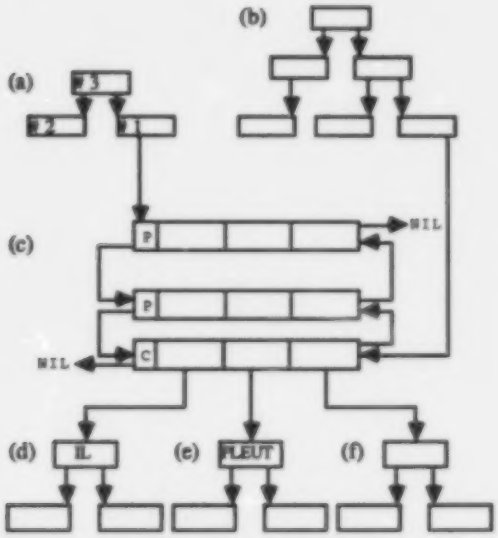


Figure 4 : Diagramme de structure de données

Liens avec la psychologie cognitive

Bien que les êtres humains n'utilisent pas constamment des modes de raisonnement fondés sur la logique, celle-ci constitue la principale forme de raisonnement de certains domaines d'activités, notamment le domaine scientifique.

Pour les personnes qui y travaillent, le raisonnement prend généralement la forme d'un problème à résoudre, sur lequel ils appliquent différentes heuristiques fonctionnant plus ou moins bien, selon la nature du problème à résoudre [Anderson, 85]. Chaque individu, durant son enfance et le début de son adolescence, développe ces heuristiques à la suite d'essais et d'erreurs constatées lors de la résolution de problèmes. Cependant ces heuristiques ne sont pas parfaites et elles se résument souvent à un raisonnement par l'absurde (*reductio ad absurdum*) ou à une mauvaise interprétation de la proposition à résoudre [Wason et Johnson-Laird, 72]. Il n'est d'ailleurs pas trop surprenant de découvrir que des enfants, qui fonctionnent à partir d'un nombre limité d'heuristiques, réussissent mieux que des adultes à résoudre correctement des problèmes grâce aux figures du *modus ponens* et du *modus tollens* [Wason et Johnson-Laird, 72].

Il existe au moins deux grandes difficultés à surmonter lors de la résolution de problèmes, [Anderson, 85 ; Wason et Johnson-Laird, 72] soit:

- l'interprétation d'une proposition conditionnelle comme étant une biconditionnelle.
- l'échec dans l'application du *modus tollens* et découlant de ce fait, la difficulté à résoudre le problème posé par la négation et la double négation.

Des expériences ont démontré que le fait d'interpréter la proposition conditionnelle comme une biconditionnelle occasionne chez le sujet la formulation d'illogismes, tels que de nier l'antécédent ou d'affirmer le conséquent. [Suppes et Hill, 64] nous suggère que cette difficulté résulte d'une tendance à inclure des quantificateurs à la proposition; en fait, le sujet transforme le syllogisme conditionnel en syllogisme catégorique. Cela lui permet d'affirmer des résultats qui, bien que parfois valides, ne le sont pas en tout temps, tels que la proposition conditionnelle le demandait.

L'échec de l'application du *modus tollens* a été démontré par plusieurs expériences, dont celle de [Rips et Marcus, 77], dans laquelle seulement 57% des sujets répondaient correctement à une question exigeant son application. Il y a plusieurs hypothèses pour tenter d'expliquer ce comportement.

Tout d'abord, le *modus tollens* n'est pas une méthode fondamentale d'inférence alors que le *modus ponens* en est une. Ensuite, son application entraîne la négation du groupe d'antécédents, ce qui ajoute une connotation émotive et requiert une opération mentale supplémentaire. Donc la négation ne ferait pas "naturellement" partie des méthodes de raisonnement logique utilisées par un individu [Wason et Johnson-Laird, 72].

Toutefois, ce qui crée beaucoup plus de difficulté chez l'individu qui tente de résoudre un problème par *modus tollens*, c'est la présence de la négation dans l'antécédent ou le conséquent ou dans les deux à la fois. Pour résoudre une proposition telle que celle de l'exemple (3), étant donné que le fait B est affirmé, l'individu doit nier $\neg A$, c'est-à-dire affirmer A. Il doit transformer la proposition de négative à affirmative, pour l'interpréter, et ensuite il doit la nier deux fois de suite.

$$\frac{\neg A \rightarrow \neg B \qquad \therefore B}{\neg\neg A \Leftrightarrow A}$$

Dans une expérience de Jonathan Evans, les sujets ne réalisaient pas qu'ils devaient affirmer l'antécédent et donnaient des réponses telles que "Je ne peux pas ne pas aller voir un film ce soir"[4] (I can't not go to the pictures tonight). En fait, sans apprentissage de la logique formelle, les sujets utilisaient spontanément le *reductio ad absurdum* [Wason et Johnson-Laird, 72]. Bien que la plupart des gens sans formation en logique ait des problèmes à appliquer le *modus tollens*, cette difficulté est surmontable sous certaines conditions. Lorsque les psychologues emploient une méthode très concrète, faisant référence à des problèmes quotidiens, la majorité des sujets (22 sur 24) répondait correctement à une question demandant l'application du *modus tollens* [Johnson-Laird, Legrenzi et Legrenzi, 72].

En somme, il serait possible que les gens qui sont affectés de ces difficultés apprennent à utiliser correctement la logique formelle par des techniques d'apprentissage, telles que celles que notre prototype **LOGIX** permet d'appliquer. Tout en offrant à l'utilisateur la possibilité de tester des règles avec les deux figures des syllogismes conditionnels, il évite les difficultés énumérées plus haut. Dans la version actuelle du prototype toutefois, seules les propositions conditionnelles sont applicables; les biconditionnelles feront partie de développe-

[4] Traduction libre.

ments futurs du système. De plus, l'usager ne peut rencontrer de difficulté avec les négations qu'impliquent le *modus tollens*, car le système fait automatiquement les inférences nécessaires; il peut malgré tout comprendre le raisonnement employé grâce à la vérification assistée [Brulé, Thérien et Tremblay, 88].

Induction par recours à l'utilisateur

Lors de raisonnement par *modus tollens*, la collaboration de l'utilisateur est demandée pour spécifier au système la bonne combinaison de fait(s) nécessaire(s) pour maintenir la valeur de vérité de la règle, étant donné la négation du groupe de conséquents. Ainsi, si on considère l'exemple 1, en notation symbolique, avec le fait C affirmé et les faits A et B indéterminés, le système ne pourrait normalement rien conclure.

$$\frac{(A \wedge B) \rightarrow \neg C \qquad \therefore C}{(\neg A \vee \neg B)}$$

Toutefois, on peut demander à l'utilisateur d'induire la conclusion en précisant, au meilleur de sa connaissance, la valeur à associer au fait A et au fait B. Il a alors le choix entre trois valeurs possibles pour chaque prémisse(s) indéterminée(s), soit (i) affirmer une prémisse qu'il considère comme la plus probable (par exemple: ¬A); (ii) nier une prémisse qu'il considère comme la moins possible (par exemple: B) et (iii) laisser la prémisse à sa valeur initiale (i.e. indéterminée). De plus, l'utilisateur qui n'a pas l'information nécessaire pour se permettre de tirer une conclusion lorsqu'un choix se présente, peut toujours tester une valeur (affirmative ou négative) sur une prémisse et ainsi voir les répercussions lors des inférences suivantes.

Ce type d'induction, que l'on qualifie d'amplifiante [Gex, 53], n'a aucun fondement logique, mais repose plutôt sur l'interprétation que les gens ont de la probabilité associée à la vérification d'une proposition. Idéalement, cette probabilité devra être calculée grâce au théorème de Baye (modifié, étant donné l'incertitude de la valeur de vérité de certains faits), car les humains ne respectent pas parfaitement le modèle Bayesien [Anderson, 85]. Diverses expériences ont démontré qu'ils sous-estimaient la probabilité d'un événement[5] ou oubliaient la probabilité des expériences précédentes. Considérant l'évolution du prototype, ce non-respect de la probabilité "exacte" ne nous a pas semblé être un obstacle, car l'utilisateur a toute la latitude pour proposer la valeur d'un fait et ainsi voir les résultats sur les inférences ultérieures et revenir en arrière si nécessaire.

Toutefois, la plus grande interrogation face au mode d'induction employé dans notre système demeure sa validité. Ainsi, selon [Skyrms, 75], une proposition peut être fortement induite si, et seulement si, elle respecte deux conditions, soit: elle a une probabilité inductive élevée et elle ne peut être inférée par raisonnement déductif. À cette réponse, nous opposons l'idée que tout raisonnement hypothétique est une forme d'induction, comme le dit [Gex, 53]: *"Dès que l'esprit formule une hypothèse pour expliquer la nature, il y a déjà induction, car l'esprit pense spontanément d'une manière universelle."*

Les limites de LOGIX

Même si nous n'avons pas introduit la notation avec parenthèses, au départ dans le but de faciliter le travail de l'utilisateur, nous constatons qu'il existe des possibilités de confusion au plan de l'interprétation; il s'agit donc pour le moment d'une limite que nous essaierons de corriger ultérieurement.

Une autre des limites concerne l'emploi de facteurs de certitudes. Le fait que notre système se limite à des valeurs comme vrai, faux et indéterminé limite l'usage qui peut être fait d'un tel système.

Enfin la contrainte qu'impose la logique des propositions au plan de la reconnaissance des formes (à moins de vouloir dépasser le niveau syntaxique d'interprétation) implique que les faits contenus dans la base de faits doivent comporter un niveau d'identité assez grand par rapport à ceux qui sont contenus dans les règles pour permettre un traitement satisfaisant.

Développements futurs

A ce titre nous avons déjà mentionné que nous voulons introduire le traitement des parenthèses dans les règles pour assurer la consistance de l'interprétation. Nous avons également mentionné antérieurement notre intention de rendre possible le traitement des biconditionnelles dans les règles.

Remerciements

Nous tenons à remercier Josée Tremblay pour son aide lors du développement de cette deuxième version du prototype, Benoît Champagne, Jean-François D'Arcy et Charles Sauvageau qui en ont élaboré la première version, ainsi que Simone Richard pour ses suggestions dans l'élaboration de cette communication.

Bibliographie

Anderson, J. R., 1985. Cognitive Psychology and its Implications. Second Edition, W. H. Freeman and Co., New-York.

Brulé, G., Thérien, L. et Tremblay, J., 1988, Le Système Expert, Manuel de Conception et Manuel de l'Usager. UQAM, non-publié.

Camarda, V. et Lévesque, G., 1988. Le raisonnement syllogistique, Rapport de recherche présenté à l'Association des Collèges du Québec. Avril.

Chenique, F., 1975. Eléments de logique classique. Dunod, Paris.

Edwards, W., 1968. Conservatism in human information processing. In B. Kleinmuntz (Ed.), Formal Representations of Human Judgement. Wiley, New-York.

Farreny, H., 1985. Les Systèmes Experts, principes et exemples. Cepadues-Edition, Toulouse.

Gex, M., 1953. Logique Formelle. Deuxième Edition, Editions du Griffon, Neuchatel.

Johnson-Laird, P. N., Legrenzi, P. et Legrenzi, M., 1972. Reasoning and a Sense of Reality. British Journal of Psychology, 63, 3, p. 395-400.

Larvet, P., 1984, Mini-système expert pour Apple II. Micro-systèmes, Décembre, p. 195-201.

Larvet, P., 1987. Systèmes experts en Turbo-Pascal pour PC et compatibles. Eyrolles, Paris.

Rips, L. J., et Marcus, S. L., 1977. Supposition and the analysis of conditional sentences. In M. A. Just & P. A. Carpenter (Eds.), Cognitive Processes in Comprehension. Lawrence Erlbaum Associates, Hillsdale, New-York.

Skyrms, B., 1975. An Introduction to Inductive Logic. Second Edition, Dickenson Publishing Co., Encino, California.

Suppes, P. et Hill, S., 1964. First course in Mathematical Logic. Blaisdell Publishing Co., Waltham, Massachusetts.

Wason, P. C. et Johnson-Laird, P. N., 1972. Psychology of Reasoning, Structure and Content. Harvard University Press, Cambridge, Massachusetts.

[5] [Edward, 68] utilise le terme "conservateur" pour qualifier cette attitude.

THE CIRCUITS INTELLIGENT TUTORING SYSTEM

R. Coulman, S. Bhuiyan, M. Mark, B. Prasad, M. Schweighardt, J. Greer, G. McCalla
Dept. of Computational Science
University of Saskatchewan
Saskatoon, SK

Circuits is an artificial intelligence-based tutoring system (ITS) which provides practice and feedback for students learning to solve series and parallel resistor problems in physics. The system is a problem-solving tutor that selects an appropriate problem for presentation to the student, analyses the student's solution based on knowledge of that student, provides feedback as necessary, and generates appropriate subsequent problems. The advice and problem generation is based upon the system's understanding of the student's current knowledge state (the student model), particularly potential misunderstandings or misconceptions. Advice is offered in an effort to correct misconceptions, and reinforcement of concepts is provided through additional practice and feedback. The system has been prototyped on a Symbolics 3640 in Common Lisp. Its graphic interface accepts input from the keyboard and/or a pointing device. A built-in calculator window is available for performing intermediate calculations and for entering numeric responses. A fully integrated help facility intelligently responds to student requests for help within the context that the help was requested.

CIRCUITS is a complete, working ITS that incorporates many artificial intelligence techniques. Because the domain is relatively small, we have been able to implement a complete system and thus observe the interaction between all of the components of an ITS. Continued development of the system is underway to enhance the student model and error diagnosticians so that more subtle misconceptions can be recognized and represented. In addition, more intelligent control of communication facilities is being added to determine when and how to interrupt the student. The system is currently being used at the ARIES Laboratory as a testbed for research into various ITS problems.

LE SYSTÈME TUTORIEL INTELLIGENT CIRCUITS

Le système CIRCUITS est un système tutoriel à intelligence artificielle (STI) qui offre aux étudiants un enseignement dirigé à base d'exercices pour résoudre des problèmes de physique dans le domaine des circuits à résistances en série et en parallèle. Le système choisit un problème approprié et le soumet à l'étudiant, analyse la solution proposée par l'étudiant en se basant sur les connaissances de celui-ci, au besoin lui vient en aide et produit de nouveaux problèmes appropriés. La production des messages de guidage et des problèmes est basée sur les connaissances actuelles de l'étudiant (modèle de l'étudiant) et tient compte particulièrement des malentendus ou des conceptions erronées possibles. Le guidage vise à corriger les conceptions erronées; le système a recours à des exercices supplémentaires et à la rétroaction pour renforcer les concepts. Le prototype du système est composé en Common Lisp et est installé sur machine Symbolics 3640. Son interface graphique peut être sollicitée à partir du clavier et/ou d'un dispositif de pointage. Elle comporte une fenêtre de calcul permettant d'effectuer des calculs intermédiaires et d'introduire des réponses numériques. Une fonction d'assistance répond intelligemment aux demandes d'assistance de l'étudiant dans les limites du contexte.

Le système CIRCUITS est un STI complet et opérationnel qui incorpore plusieurs techniques d'intelligence artificielle. Vu l'étendue relativement restreinte du domaine, nous avons pu mettre au point un système complet et observer l'interaction entre tous les composants du STI. Le développement du système se poursuit; on cherche à améliorer le modèle de l'étudiant et le diagnostic des erreurs afin de pouvoir détecter et représenter des conceptions erronées plus subtiles. De plus, un dispositif de contrôle des fonctions de communication plus intelligent visant à déterminer quand et comment interrompre l'élève est en voie d'être incorporé au système. Celui-ci est présentement utilisé comme banc d'essai au laboratoire ARIES pour fins de recherche dans divers problèmes reliés aux STI.

The CircuITS Intelligent Tutoring System

R. Coulman, S. Bhuiyan, M. Mark, B. Prasad
M. Schweighardt, J. Greer and G. McCalla

ARIES Laboratory
Department of Computational Science
University of Saskatchewan
Saskatoon, Canada S7N 0W0

Abstract

CircuITS is an artificial intelligence-based tutoring system (ITS) which provides practice and feedback for students learning to solve series and parallel resistor problems in physics. The system is a problem-solving tutor that selects an appropriate problem for presentation to the student, analyzes the student's solution based on knowledge of that student, provides feedback as necessary, and generates appropriate subsequent problems. The advice and problem generation is based upon the system's understanding of the student's current knowledge state (the student model), particularly potential misunderstandings or misconceptions. Advice is offered in an effort to correct misconceptions and reinforcement of concepts is provided through additional practice and feedback. The system is currently being used at the ARIES Laboratory as a testbed for research into various ITS problems.

1. Introduction

Intelligent tutoring systems provide students with an individualized learning environment and some of the subtlety and flexibility of a knowledgeable tutor, coach, or guide. The issues underlying the construction of intelligent educational systems are currently the subject of intense research. Several collections of papers have been produced (e.g. Sleeman and Brown, 1982; Kearsley, 1987; Psotka et al., 1988) and at least one textbook (Wenger, 1987) has been written, in an effort to try to integrate the many diverse threads of this multi-disciplinary effort. These research contributions indicate that most systems focus on a single aspect and that complete, operational, real-time intelligent tutoring systems (ITS) are relatively rare. The CircuITS system described in this paper represents an instance of a complete, real-time ITS, which makes use of many artificial intelligence techniques to achieve competent tutoring performance.

CircuITS is an intelligent problem-solving environment designed to assist students in acquiring the skill to solve equivalent resistance problems in electrical circuits. Complex aggregations of resistors in series and parallel circuits can be "simplified" to an equivalent single resistor, and thus the overall resistance in the circuit can be computed. This skill comprises the ability to solve simple parallel and series resistors, as well as the ability to appropriately decompose complex resistor aggregations into simple series or parallel groups. The CircuITS tutor originated as a project in an Artificial Intelligence graduate course at the University of Saskatchewan and was subsequently enhanced to its current state.

The strategy of the CircuITS tutor is to begin with a simple problem, to track the behaviour of the student trying to solve the problem, to diagnose the student's current knowledge state, and to provide relevant feedback. Subsequent problems specifically tailored to the student's needs are then designed automatically by the tutoring system. The tutor tracks the student's past actions and diagnoses possible misconceptions by recognizing patterns in the student's behaviour. As the tutor accumulates more knowledge about the student, more appropriate feedback can be designed.

2. The CircuITS System Architecture

The architecture of intelligent tutoring systems differs significantly from traditional computer-assisted learning systems. Typically, an ITS contains six interacting components: 1) a User Interface which permits communication between the student and the ITS, 2) a Domain Expert that knows about the subject matter to be taught and is capable of solving problems in the domain, 3) a Diagnostic Component that detects student errors and understands misconceptions associated with error patterns, 4) a Tutoring Specialist that can plan instruction and feedback, 5) a Student Model that maintains and monitors the systems beliefs about the student's knowledge state, and 6) a Control Unit that coordinates the interaction of the other components.

Due to the nature of the electrical circuits tutoring domain, the development of CircuITS has concentrated on each of these six components to a varying degree. Since the domain is narrowly defined, the domain expert has been trivialized with domain expertise built into each of the other components on an as-needed basis. The Tutoring Specialist takes a central role in this system, and as such has been subdivided into two components, the Instructional Planner and the Feedback Component. Each of the sub-sections below describes design decisions and attributes of the various components of CircuITS.

2.1 User Interface

The user interface for CircuITS consists of five interactive windows: the command menu, the calculator, the circuit window, the help window, and the instruction window. Figure 1 illustrates the screen layout. The user interface was designed so that students could flexibly interact with the tutor by using a pointing device (mouse), the keyboard, or a combination of the two.

The command menu (in the lower left corner) enables the student to start (or restart) the tutor or to request general help on the use of the tutor. The calculator window enables the student to perform intermediate calculations and to enter numeric responses. It is a standard four-function calculator with parentheses for enforcing evaluation order and a reciprocal (1/x) key. The calculator also offers other functions for communicating with the tutor, such as the Continue and Enter buttons used to dispatch various inputs to the tutor. Full natural language input was not incorporated into CircuITS because the typical input from the student is a number, a symbol, or a command. Such inputs are adequately handled through pointing devices and menu or button-oriented input facilities.

The instruction window is used to interact with the student. It displays prompts from the tutor directing or coaching the student through a problem solution. In addition, all explanations of errors and other feedback messages appear in this window. The tutor guides the student through the solution to each problem, providing prompts in the instruction window at each step. If the student is unsure of what to do next, the help button summons additional information.

In the circuit window (lower right corner) the resistor problem to be solved is graphically displayed for the student. The circuit window also contains several "buttons" for use in identifying the components of the circuit that the student wishes to simplify next. The circuit window is capable of displaying problems of arbitrary complexity, limited only by the size of the window. Given the current size, it is possible to pose problems with up to sixteen resistors and six levels of nesting.

If the student requests help in solving a problem at any point, the help window is used to display messages. These messages are "context sensitive" in that the system provides help which corresponds to the student's current activity.

2.1.1 An Example

In order to solve the problems that CircuITS generates, there are two main steps: 1) the identification of a sub-component of the aggregate resistor set to solve and 2) the calculation of an equivalent resistance. The interface is designed so that each of these functions occurs in the lower area of the screen. A pop-up menu in the circuit window is used to identify the sub-component and the calculator window is used for calculating and/or entering an equivalent resistance.

Figure 1 shows an example of a session already in progress. The problem displayed in the Circuit Window has just been given to the student. In the Help Window, the tutor explains to the student that the goal is to reduce the number of resistors to a smaller number with an equivalent net resistance and to continue reductions until only a single resistor remains. Explicit direction regarding the immediate next step is also provided in the Instruction Window.

A student's solution to a problem is composed of a series of identification and simplification steps. A group of resistors is identified for simplification and then the group is simplified to a single resistance. During identification, the student selects a set of resistors (a sub-component) and states whether resistors in that sub-component are in series or parallel. (Currently sub-components containing a mixture of series and parallel resistors may not be selected for a single simplification step). At this stage, the student might select resistors **R3** and **R15** (in Figure 1) and state that they are in parallel. Next the student performs a calculation to compute the resistance equivalent to that of the selected sub-component. This calculation may be explicitly entered into the calculator or, if desired, any numeric value or expression may be submitted to the tutor. For example, suppose the student enters the following expression into the calculator: $(22^{-1} + 42^{-1})^{-1}$. At this point, the student has given a solution to part of the problem. The tutor now takes this information and tries to infer what the student has done and how to respond to the student. Other sections of this paper describe how this inference is done.

If the student's response were correct, the circuit would be re-drawn with a single new resistor replacing the sub-component that was simplified. (A simplification of the Figure 1 circuit is shown in Figure 3). The student then would proceed with further simplification of the resistors until only a single resistor (equivalent to the initial set of resistors) remains. If the student's response is incorrect, feedback appropriate to the student's level of understanding is designed. Regardless of the student's response, a great deal of intelligent processing is performed by the tutoring system as it decides what to do next.

2.2 Diagnostician

The diagnostic component of CircuITS embodies more "intelligence" than any other component of the system. It determines the student's problem-solving strategy and decides where flaws might exist in the strategy or in carrying out the strategy.

The first task of the diagnostician is to examine the student's input and to deduce what the student is trying to accomplish. The student's identification of the resistor sub-component is compared with the actual relationship between the identified resistors. There are four possibilities for this actual relationship: series, parallel, adjacent, and non-adjacent. For example, in Figure 1, resistors **R1** and **R2** are in series, while **R10** and **R13** are in parallel. Resistors **R5** and **R10** are neither in parallel nor in series and are said to be "adjacent". Finally, resistors **R5** and **R13** are neither in parallel nor in series nor adjacent, and are said to be "non-adjacent".

Inference techniques are used to determine the formula used by the student in the calculation. Since the mathematical expression used by the student may not have been explicitly entered into the calculator, the formula often must be inferred from the result. Plausible candidate formulas that could generate the student's response are proposed.

After this first level of diagnosis, the identification and calculation are combined to determine whether or not the calculation was correct. Evidence from multiple sources including the student's identification of the sub-component, the actual relation of resistors in the sub-component, the formula and its result, and prior knowledge of the student (described below) is considered in making inferences about the student's intentions and current knowledge state.

2.3 Student Model

The student model is the component of an ITS that represents the system's knowledge about the student's knowledge. Inferences about student knowledge made by the diagnostician are thus

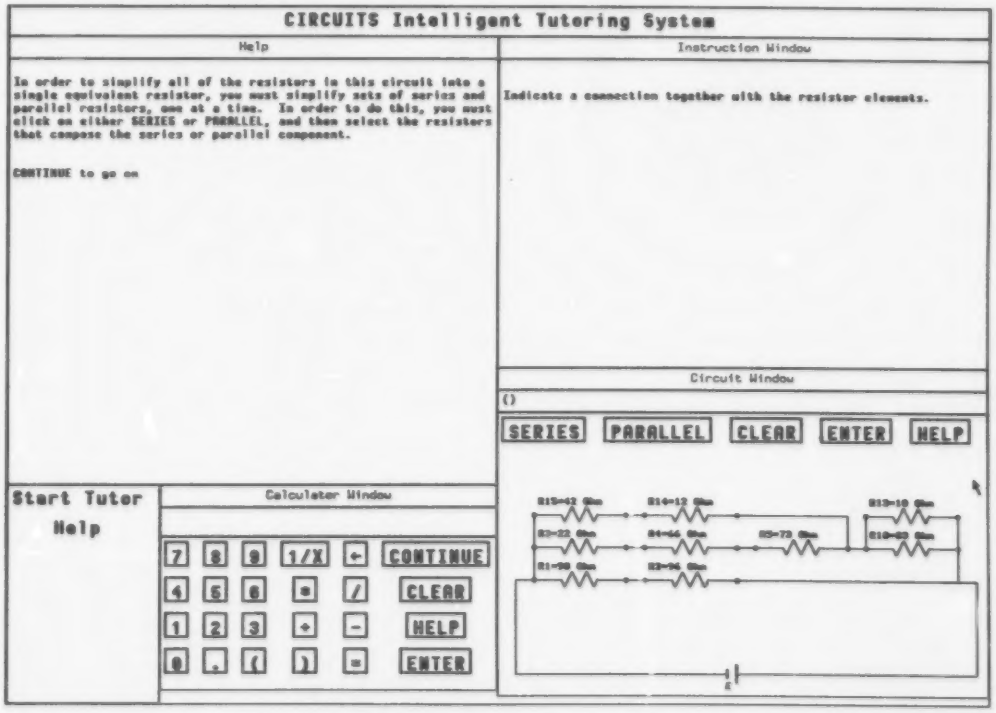


Figure 1. The CircuITS Display with a Sample Problem

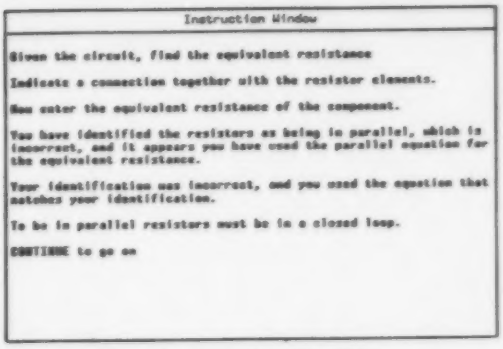


Figure 2. Responding to an Error

maintained by the student model. The system's knowledge about the student's knowledge is uncertain; in fact, the system can at best possess certain beliefs or hypotheses about the student's knowledge state.

It must be realized that a student can make errors or may exhibit non-ideal behaviours for various reasons. First of all, an error may be an accidental slip, such as pressing the wrong key. On the other hand, the student might not understand a particular sub-concept and might consistently make mistakes related to that sub-concept. The student might have a serious misconception that leads to certain patterns of behaviour detectable only over longer periods of time. The student model must be able to compensate for slips in order that the student not be heavily penalized for these mistakes. It must also record the frequency with which the student makes the same types of errors and how each error relates to the desired concepts.

In CircuITS, the student model is implemented using a variation of the genetic graph (Goldstein, 1982; Brecht and Jones, 1988). The student model is composed of three levels of abstraction. At the lowest level, raw information about the student's actions is stored. This information includes 1) what circuit component identification was made, and 2) what formula was most likely used.

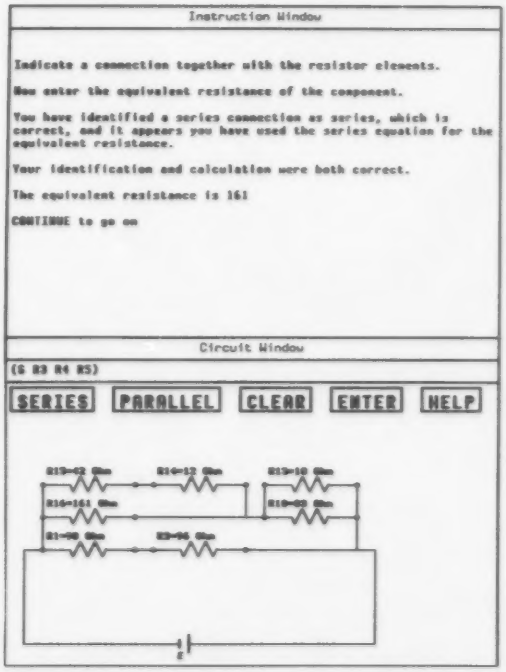


Figure 3. Responding to a Correct Simplification

Using these low-level concepts, the diagnostician makes inferences about the correctness of a student response and the appropriateness of the formula used by the student. These inferences are stored at the second level of the student model. Note that first level information is insufficient to determine whether or not the student used the correct formula. For example, the student may have used the series equation, but unless the components he or she identified were actually in series, this calculation represents an error.

As the student solves more problems, certain patterns in responses may begin to appear. These patterns, if interpreted correctly, can go beyond the analysis of isolated errors and can reveal potential deeper misconceptions. The third level of the student model is used to store information regarding these misconceptions. More sophisticated recognition at this level remains to be accomplished. Currently CircuITS performs only limited diagnosis at this third level; however, preliminary results are encouraging.

In CircuITS, the ideal state of the student model contains all concepts that the student will need to "know" in the domain of series and parallel circuits. At the lowest level, there are four behaviours that need to be recognized. These behaviours correspond exactly to the first level of diagnosis by the diagnostic component. Thus, it is necessary for the student to be able to use both the series and the parallel equations, and to be able to identify series and parallel components as such. These first level behaviours combine to yield the second level abstractions. That is, appropriately identifying a series/parallel component as series/parallel and using the series/parallel equation on that component implies that the student understands series/parallel circuits. The third level abstractions roughly correspond to understanding both series and parallel circuits and related misconceptions.

The student model is completely updated each time the student carries out a solution step. The student model is utilized in selecting each subsequent problem for the student to solve, for designing feedback appropriate to the particular student and for diagnosing misconceptions. During each update, the goal concepts (in the ideal student model) are analyzed in combination with errors that student has made related to each concept. This analysis determines whether it is safe to assume that the student knows the concept. If the student made several errors of the same type, but has since made several correct responses in the same context, then the more recent correct answers out-weigh the earlier errors.

The student model has been designed so that there is room for expanding the diagnostic capabilities of the tutor without having to redesign the student model. It is a very flexible component and embodies a reasonable amount of subtlety.

2.4 Planner

The planning component of CircuITS is designed to use information provided by the student model, to determine what type of problem to generate next and to design a particular problem of that type. In order to accomplish this, several requirements must be considered. First of all, it is necessary to group problems together into levels of difficulty. In CircuITS, there are seven levels of difficulty. Level zero is a simple two resistor component. Level one is a simple component of two to four resistors and levels two through six correspond to the number of levels of nesting in a given problem. The maximum of six levels of nesting (and corresponding levels of difficulty) is constrained only by the current size of the circuit display window. This is not a serious constraint, as we have determined that students who can solve equivalent resistor problems for circuits with three or four levels of nesting can solve arbitrarily complex nested circuits. For each level of nesting there are two general classes of problems: those with their largest resistor clusters connected in series and those with their largest resistor clusters connected in parallel. There exists a strict partial ordering of problem types such that a prerequisite structure of problem types may be created. Therefore the criterion may be imposed that a student must be able to solve all prerequisite problem types before being allowed to attempt a given problem. In this way, the student is not challenged to solve a problem far beyond his or her capability. At the same time, in the absence of negative evidence, one successful solution to a problem of a particular type is sufficient to demonstrate "tentative mastery" and the student may progress. In this way, students may proceed rapidly up to their level of competence.

In order to select a valid problem type, the planner queries the

student model to determine which problem types the student has successfully solved. If all prerequisites for any new problem types have been met, then this new problem type is added to a list of valid candidate problem types. Using this list, a problem type is randomly selected and an actual problem is generated.

The generation of problems follows several simple rules and constraints. The display constrains the maximum number of resistors, the problem type indicates the top-level connection, and the desired number of levels of nesting constrains the configuration of the resistors. Finally, a configuration of resistors can be simplified using the following rules: several series components is the same as a single series component, and several parallel components in parallel are the same as a single parallel component. That is, only parallel components and single resistors can be in series and only series components and single resistors can be in parallel.

2.5 Feedback

Although an ideal system for feedback to the student would probably be based on the ability to interpret and generate natural language, this is outside the scope of the CircuITS project. CircuITS uses a fairly simple, yet effective strategy for providing feedback to the student. The first level diagnostics are reported directly to the student using prestored natural language sentence templates. This includes the component identification made by the student, the actual relationship within that component, and the formula that the tutor believed the student used. The second level diagnostics are also reported. If the student has made an error, the tutor explains the nature of the error and the possible cause of the error. It then tries to give advice in the form of a correct formula or an explanation of what a parallel connection looks like, etc. Figure 2 shows the feedback given as a result of the student's incorrect response given to the problem in Figure 1 (described in section 2.1.1). The error in the response was that the identified resistors were not actually in parallel with each other. Rather, they are first sub-components of two series components which are in parallel with each other.

The student's answers reflected this belief and the tutor was able to deduce this and provided a definition of parallel circuits to aid the student in understanding the concept. Following this feedback, the student would be given another chance to solve the problem correctly. For example, if the student were to simplify the series component consisting of **R3, R4, and R5** (Figure 1) and were to give the correct equivalent resistance, the tutor's response is shown in Figure 3. The tutor acknowledges the correct answer and then simplifies the component, re-drawing the circuit with the equivalent resistor **R16** replacing **R3, R4, and R5**. The student would then continue with this problem until it was completely solved.

Current development is focussing on the third level of the student model and diagnosing student misconceptions. An interesting aspect of this area is providing feedback to the student based on these misconceptions. In order to correct misconceptions, more subtle interaction with the student and more ingenuity in instructing the student are required. It is here that a full-fledged natural language interface would help the most. Different methods of instruction can be used here, based on past interaction with the student. For example, some students may learn best by analogy. Others may prefer generalization of concepts they already know. Once the method of instruction has been decided, based on information in the student model, the feedback component would then explain to the student what the misconception is and what is wrong with it. Then, using the desired method, the tutor would give the student further instruction and information in order to understand the concept. Another problem would then be given to allow the student a chance to practice what was just explained. The tutor would also be able to determine if the student learned anything from the lesson. If not, it would be necessary to try a different method of instruction.

2.6 Control

The control unit coordinates all the activities of the tutor. Each of the other components of CircuITS relies on messages communicated via the control unit. A general tutoring cycle is established in which a problem is generated and displayed, a response is obtained and analyzed, the student model is updated, feedback is designed and delivered, and an appropriate follow-up action is taken. Certain activities of the system, however, are more or less independent of this cycle. Inferences to detect misconceptions using the student model knowledge base, plans for specialized instruction to remove misconceptions, and plans for timing interruptions of the tutoring cycle proceed (effectively) in parallel with the standard tutoring cycle. After a planned interruption of the tutoring cycle has been completed, the cycle may be resumed or continued from a new point.

3. Conclusions and Future Directions

There are several areas in which to enhance CircuITS. The first and most important involves developing the third level of the student model and implementing the inference component in order to be able to diagnose misconceptions.

Another idea is to add a hinting mechanism to the help system, so that a student can ask for a hint when stuck on a problem. Also, the tutor would be able to give the student a hint to solve the problem, if it appeared the student was having difficulty.

Another desirable feature is to incorporate preferred teaching methods into the student model. If the student learns well with one type of instruction, but has trouble with others, then the tutor should use the method that would seem likely to work the best.

Another possibility is the idea of "dynamic problems". If the student is having trouble with a given component of a problem, suspend solution of the problem and give a problem which contains only the component in question. The student will already have seen a problem of this type, and should be able to solve it. After the smaller problem has been solved, the larger problem can be given again and the student can then make analogies which should help in solving the problem. This is similar to the approach taken in the BIP Programming Advisor (Westcourt, Beard and Gould, 1977)

CircuITS is a complete, operational, real-time intelligent tutoring system. It is not finished, nor perfect, but all of its components are fully functional. Several ITS components have been incorporated into a single environment, and the resulting system has now become a testbed for further study. With this system it is now possible to observe how students interact with the tutor and to begin evaluating the effects of this system and of ITS's in general.

4. References

Brecht, B. and Jones, M. Student Models: The Genetic Graph Approach, The International Journal of Man Machine Studies, 1988.

Goldstein, I. P. The genetic graph: a representation for the evolution of procedural knowledge, in D. Sleeman and J. S. Brown (Eds) Intelligent Tutoring Systems, London: Academic Press, 1982, 51-78.

Kearsley, G. P. Artificial Intelligence and Instruction: Applications and Methods, Don Mills, Ontario: Addison Wesley, 1987.

Psotka, J. Massey, L. and Mutter, S. Intelligent Tutoring Systems: Lessons Learned, Hillsdale, NJ: Lawrence Erlbaum, 1988.

Sleeman, D. and Brown J. S. (Eds) Intelligent Tutoring Systems, London: Academic Press, 1982.

Wenger, E. Artificial Intelligence and Tutoring Systems,. Los Altos, CA: Morgan Kaufmann, 1987.

Westcourt, K., Beard, M. and Gould, L. Knowledge-Based Adaptive Curriculum Sequencing for CAI, Proc. ACM National Conference, Seattle, Washington, 1977, pp. 234-240.

VERS UN SYSTÈME TUTORIEL INTELLIGENT ENSEIGNANT UNE STRATÉGIE POUR L'INTÉGRATION MATHÉMATIQUE

H. Beaulieu
Laboratoire d'intelligence artificielle en éducation
Université Laval
Québec, PQ

A. Baronian, J.-C. Bélanger
Collège de Sainte-Foy
Sainte-Foy, PQ

Dans cet article, nous présentons un système tutoriel intelligent qui enseigne une stratégie de contrôle dans le domaine de l'intégration mathématique. Cette stratégie prescriptive, simulant la démarche de l'expert, a pour but d'aider l'étudiant dans la sélection d'une technique particulière d'intégration.

Notre système, présentement en développement, est mun d'une base de connaissances à deux niveaux: un niveau expert relié aux démarches les plus appropriées et un deuxième niveau relié aux démarches succeptibles de causer certains difficultés. Le modèle étudiant, quant à lui, se construit au fur et à mesure de l'intéraction et permet d'adapter les interventions aux besoins de chaque étudiant. Finalement, les connaissances pédagogiques du système sont présentées sous forme de règles dans un module tutoriel.

TOWARDS AN INTELLIGENT TUTORING SYSTEM FOR TEACHING STRATEGY FOR MATHEMATICAL INTEGRATION

In this article, we will present an intelligent tutoring system that teaches a control strategy in the field of mathematical integration. The purpose of this prescriptive strategy, simulating the expert's procedure, is to assist the student in selecting a specific integration technique.

Our system, which is currently under development, has a knowledge base at two levels: an expert level related to the most appropriate procedure and a second level involving steps likely to cause certain difficulties. The student profile is formed as interaction progresses and allows adaptation of the operations to the needs of each student. Finally, the educational knowledge of the system will be presented in the form of rules in a tutoring module.

VERS UN SYSTEME TUTORIEL INTELLIGENT ENSEIGNANT UNE STRATEGIE POUR L'INTEGRATION MATHEMATIQUE

Hélène Beaulieu[1], Arménouhie Baronian[2] et Jean-Claude Bélanger[2]

[1]Laboratoire d'Intelligence Artificielle en Education
Université Laval, Québec
[2]Collège de Sainte-Foy, Québec

Résumé

SIM est un système tutoriel intelligent qui enseigne une stratégie dans le domaine de l'intégration mathématique. Cette stratégie prescriptive, simulant la démarche de l'expert, a pour but d'aider l'étudiant lors de la sélection d'une technique particulière d'intégration.

Présentement en développement, **SIM** possède des connaissances de niveau expert reliées aux démarches les plus appropriées lors de l'intégration d'une fonction et des connaissances reliées aux démarches susceptibles de causer certaines difficultés. Les démarches de l'étudiant sont enregistrées dans un module permettant d'évaluer son niveau de performance et de sélectionner des stratégies d'interventions adaptées à ses besoins.

1. Introduction

La plupart des systèmes tutoriels existants ont été développés pour explorer les capacités des techniques de l'intelligence artificielle dans les processus éducatifs (Park, Perez et Seidel, 1987). Plusieurs d'entre eux ont contribué à l'amélioration de la compréhension des processus cognitifs impliqués dans l'apprentissage et l'enseignement de tâches spécifiques Cependant leur portée quant à la valeur pratique en tant que systèmes pédagogiques reste encore très limitée.

Park, Perez et Seidel expliquent ce phénomène par le peu d'attention apportée aux questions pédagogiques importantes durant le processus de développement. A cet égard, ils proposent de tenir compte de certains aspects lors du développement de systèmes tutoriels intelligents, dont les suivants:

- l'application des techniques de l'intelligence artificielle au développement des systèmes tutoriels intelligents devrait se concentrer beaucoup plus sur l'efficacité pédagogique; les techniques de l'intelligence artificielle devraient être choisies (ou modifiées) en fonction des besoins pédagogiques;

- les systèmes à venir devraient mettre plus d'emphase sur le processus prescriptif du système en intégrant différents principes et stratégies pédagogiques (dépendants ou indépendants de la matière enseignée).

Le projet **SIM** a été conçu dans cet esprit. Nous avons d'abord concentré nos efforts à essayer de cerner les difficultés qu'éprouvent les étudiants lorsqu'ils résolvent des intégrales; nous nous sommes inspirés des travaux de Schoenfeld (1978, 1985) nous permettant d'entrevoir une approche pédagogique différente. Cette dernière a d'ailleurs servi de moteur à la conception et au développement de **SIM**.

2. Notre approche informatique

Le temps pour développer un système tutoriel intelligent, la maîtrise des techniques de l'intelligence artificielle et de programmation et le type d'équipement requis sont tous des facteurs qui semblent faire obstacle à une plus grande utilisation des systèmes tutoriels intelligents (Begg, Hogg, 1987). Pour rendre plus accessible l'utilisation de ces systèmes en milieu scolaire, il nous apparaît indispensable de faire des réductions substantielles au niveau de ces trois facteurs. Différentes approches nous sont alors offertes dont:

- l'utilisation d'un système auteur intelligent, c'est-à-dire un système qui génère un système tutoriel intelligent;

- l'utilisation d'un système auteur traditionnel, enrichi d'un certain nombre de modules visant à lui donner les connaissances indispensables pour atteindre le niveau d'un système tutoriel intelligent.

Pour le développement de notre système, nous avons porté notre choix sur cette deuxième approche qui s'appuie sur trois idées. La première est que la structure des systèmes actuels convient bien à la transmission d'informations. La seconde constate que les questions qui ne nécessitent pas plus d'une étape peuvent continuer à être contrôler par un processus classique d'analyse de réponses. La dernière stipule qu'au sein de la structure précédente, il peut être aisé de faire appel à des modules capables de raisonnements complexes destinés à la résolution de problèmes (Regourd, 1988).

La perspective d'atteindre des résultats positifs et tangibles assez rapidement et le désir de développer notre système sur micro-ordinateur sont aussi des facteurs qui ont motivé ce choix. Le projet **SIM** se veut donc une exploration des applications possibles de cette approche au développement de systèmes tutoriels intelligents.

3. Une stratégie pour l'intégration mathématique

La métacognition réfère à la connaissance et au contrôle que nous avons sur nos processus cognitifs. La métacognition couvre deux aspects : les connaissances métacognitives et les conduites métacognitives. Ce dernier aspect de la métacognition concerne les conduites qu'une personne peut manifester durant la réalisation d'une tâche cognitive, en vue d'utiliser les ressources dont elle dispose avec un certain degré d'efficacité (Flavell, 1976). Brown et al. (1983) donne une description intéressante de ces conduites.

> The processes include planning activities prior to undertaking a problem (predicting outcomes, scheduling strategies, various form of vacarious trial and error, etc.), monitoring activities during learning (testing, revising, reschudeling one's strategies for learning), and checking outcomes (evaluating the outcome of any stratégic actions against criteria of efficiency and effectiveness). (p. 107).

Dans le domaine des mathématiques, Schoenfeld a effectué plusieurs études sur les conduites métacognitives. Ces études l'ont amené à croire que le manque de succès de l'enseignement dans le domaine de la résolution de problèmes mathématiques serait principalement dû au fait que l'on accorde peu d'attention aux conduites métacognitives. Il a entre autre mené une étude (1978) pour décrire l'impact de certaines de ces conduites sur la performance d'étudiants de niveau pré-universitaire (collégial) dans le domaine de l'intégration mathématique.

A cette fin, il a élaboré une stratégie qu'il a appelé stratégie de contrôle (le terme contrôle fait ici référence aux conduites métacognitives relatives à l'allocation des ressources telles que les procédures, les algorithmes, les heuristiques,...) en s'inspirant des techniques développées en intelligence artificielle suite à l'observation d'experts lors du processus de résolution de problèmes. Cette stratégie, présentée à la figure1, simule la démarche de l'expert lors de la sélection d'une technique particulière d'intégration.

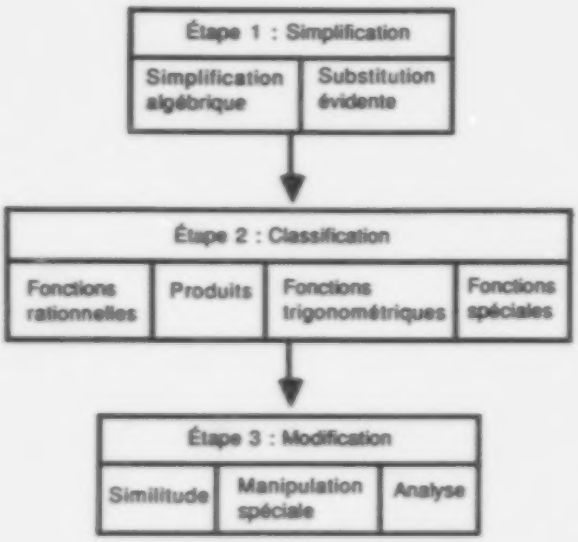


Figure 1

Cette stratégie fut l'objet d'une expérimentation auprès d'étudiants suivant un cours de calcul élémentaire et les conclusions furent les suivantes:

- les étudiants ne sont pas capables de développer des stratégies efficaces de contrôle et leur performance au niveau de la résolution d'intégrales indéfinies en est affectée;
- une stratégie de contrôle prescriptive comme celle qui est présentée plus haut, influence significativement la performance des étudiants (meilleurs résultats en moins de temps).

4. Le didacticiel SIM

Dans les cours de calcul élémentaire, on a souvent tendance à penser que la réussite de l'étudiant est proportionnelle au nombre d'exercices qu'il effectue. De cette façon, il arrive à développer des mécanismes efficaces pour la sélection d'une technique d'intégration appropriée. Or, on constate que

- la plupart des étudiants performent assez bien au niveau de l'utilisation de chacune des techniques d'intégration (substitutions, par parties, fractions partielles,...) en autant qu'on leur indique la technique à utiliser (par exemple, lorsqu'ils travaillent à la fin d'une section particulière d'un livre).

Par contre, les étudiants ont de la difficulté à sélectionner la technique d'intégration "adéquate" face à un problème donné lorsqu'ils sont en dehors d'un contexte particulier (par exemple, lorsqu'on leur propose des exercices mélangés ou lors d'un examen).

Cependant, les travaux de Schoenfeld nous permettent d'envisager qu'il est possible d'améliorer les performances des étudiants dans le domaine de l'intégration mathématique. Tel est l'objectif du projet **SIM**.

SIM est un outil d'aide aux étudiants leur permettant d'acquérir une stratégie pour les aider à sélectionner plus facilement la technique d'intégration appropriée lors de la résolution d'une intégrale indéfinie. La stratégie enseignée se présente en deux étapes correspondant aux deux premières étapes de la stratégie élaborée par Schoenfeld (figure 1). Il s'adresse à des étudiants de niveau collégial ou pré-universitaire qui suivent un premier cours de calcul différentiel et intégral.

SIM est un système tutoriel de type aviseur didactique. Il observe l'étudiant dans sa tâche de sélection d'une technique d'intégration, en déduit les opportunités d'aide et active cette aide selon certaines règles pédagogiques bien précises. Dans ce type de système, le rôle de "l'aviseur" est d'observer les décisions de l'étudiant lorsqu'il accomplit une tâche et d'intervenir au besoin pour le conseiller. De cette façon, l'étudiant est amené graduellement vers l'apprentissage des différentes composantes de sa tâche qui consiste ici, à sélectionner une technique d'intégration. Cette approche permet à l'étudiant de cheminer selon son rythme et ses besoins.

SIM possède deux types de connaissances: des connaissances de niveau expert reliées aux démarches les plus appropriées lors de l'intégration d'une fonction et des connaissances reliées aux démarches susceptibles de causer certaines difficultés. Certaines de ces connaissances permettent à l'étudiant de confronter sa démarche à celle d'un "expert" lors de la sélection d'une technique d'intégration. Dû à la structure de notre approche informatique, la base de connaissances de **SIM** ne peut cependant être définie comme une base traditionnelle distincte que l'on retrouve dans la plupart des systèmes tutoriels intelligents.

SIM fonctionne sur Macintosh et est développé à partir du logiciel Course Builder, un système visuel de développement de didacticiels polyvalent et simple à utiliser. Il incorpore un grand nombre de fonctions qui facilitent la création de didacticiels. D'une part, il permet d'intégrer des graphiques ou des textes produits avec d'autres outils du Macintosh comme Macpaint, Macdraw, Mathwriter, ect... D'autre part, le système de développement comprend un ensemble d'icônes pour construire les blocs et les enchaînements du cours ainsi qu'une structure hiérarchique pour l'organisation plus générale. Finalement, il est possible de faire de l'animation, de synthétiser la parole avec Maintalk et de produire une sortie musicale avec Musiworks. Les didacticiels construits sont autonomes c'est-à-dire qu'il n'est pas nécessaire de charger Course Builder au moment de leur exécution.

5. Conclusion

Le projet **SIM** a pour objet l'élaboration d'un système tutoriel intelligent destiné à améliorer la performance des étudiants dans le domaine de l'intégration mathématique. Son intérêt se situe à deux niveaux. D'une part, il propose une approche pédagogique différente de celle que l'on retrouve habituellement dans le milieu scolaire. D'autre part, il explore une situation informatique qui tient compte des ressources disponibles, des besoins pédagogiques et des applications des techniques de l'intelligence artificielle.

Bien que **SIM** soit encore en développement, nous croyons qu'il suscitera un intérêt autant chez les étudiants qui pourront l'utiliser comme outil d'aide à l'apprentissage que chez les enseignants qui trouveront en lui un support à l'enseignement.

6. Bibliographie

Begg, I. M. et Hogg, I., (1987). Autoring Systems for ICAI. In Greg Kearsley (Ed.). Artificial intelligence and Instruction: Applications and Methods. Addison-Wesley Plubishing Inc.

Brown, A.L. et al., (1983). Learning remembering and understanding. In P. H. Mussen (Ed.). Handbook of child psychology. (Vol. 3). New York, Wiley.

Flavell, J.H., (1976). Metacognitive aspects of problem solving. In L. Resnick (Ed.). The nature of intelligence. Hillsdale, N.J., Erlbaum.

Park, o., Perez, R.S. et Seidel, R.J., (1987). Intelligent ICA: Old Wine in New Bottle or New Vantage. In Greg Kearsley (Ed.). Artificial intelligence and Instruction: Applications and Methods. Addison-Wesley Plubishing Inc.

Schoenfeld, A.H., (1985). Mathematical problem solving. New York, Academic Press.

Schoenfeld, A.H., (1978). Presenting a strategie for indefinite integration. In American Mathematical Monthly. 85(8). p 673-678.

Regourd, J.P., (1988). Définition et implémentation d'un niveau méta en E.I.A.O. Communication présentée à la Conférence ITS-88, Montréal.

NAIADE, UN SYSTÈME EXPERT POUR DIALOGUER AVEC L'ÉLÈVE

M. Joab
Lab. d'informatique fondamentale
Université Paris 6
Paris, France

E. Cauzinille-Marmeche
Lab. de psychologie du développement et de l'éducation de l'enfant
Université Paris 5
Paris, France

J. Mathieu
Lab. de psychologie et d'informatique de Rouen
LAFORIA-CNRS, Université Paris 6
Rouen, France

NAIADE, environnement d'apprentissage d'aide au calcul algebrique, utilise les techniques les plus recentes d'Intelligence Artificielle. Il repose sur le dialogue avec l'élève, et en particulier sur l'explication. Pour atteindre notre objectif de dialogue explicatif, nous avons dû construire une organisation des connaissances originale sur trois aspects:

1) La formalisation des métaconnaissances: Les heuristiques de nature stratégique (comment orienter la stratégie de résolution?, dans quel ordre examiner les règles pour parvenir à la méthode la plus économique?..) sont explicitées et justifiées.

2) L'expression des connaissances à différents niveaux de décomposition: Les étapes de la résolution des diverses tâches (factoriser, résoudre une équation..) peuvent être plus ou moins détaillées par le système, chaque niveau de décomposition renvoyant à des concepts mathématiques distincts.

3) Dans la formalisation du réseau sémantique, les connaissances conceptuelles sont reliées aux savoir-faire procéduraux et stratégiques.

La modèlisation de connaissance experte dans le système NAIADE (généralisable à divers domaines de connaissances) contient donc en elle-même la possibilité d'explications diversifiées et adaptées au profil cognitif des utilisateurs.

NAIADE, AN EXPERT SYSTEM FOR INTERACTING WITH THE STUDENT

NAIADE, a learning environment to assist in algebraic calculation, uses the latest techniques of artificial intelligence. It is based on interaction with the student, particularly explanation. To achieve our objective of explanatory interaction, we had to prepare an original organization of knowledge in terms of three aspects:

1) The formalization of metaknowledge: The strategic heuristics (how to orient the resolution strategy, what order to follow in examining the rules to find the most economical method) are explained and justified;

2) The expression of knowledge at different levels of breakdown: The stages in the resolution of various tasks (factorizing, solving an equation and so on) can be detailed to a certain extent by the system, with each level of breakdown referring to distinct mathematical concepts;

3) In the formalization of the semantic network, the conceptual knowledge is linked to the procedural and strategic know-how.

Modelling of the expert knowledge in the NAIADE system (which can be generalized to various fields of knowledge) thus inherently involves the potential for diversified explanations adapted to the cognitive profile of users.

NAIADE, UN SYSTEME EXPERT POUR DIALOGUER AVEC L'ELEVE

Michelle JOAB 1, Evelyne CAUZINILLE-MARMECHE 2, Jacques MATHIEU 3

1 Laboratoire d'Informatique Fondamentale, PARIS 6,
2 Laboratoire de Psychologie du Développement et de l'Education de l'Enfant, PARIS 5,
3 Laboratoire de Psychologie et d'Informatique de ROUEN, LAFORIA-C.N.R.S., PARIS 6.

Résumé

NAIADE, environnement d'apprentissage pour le calcul algébrique, utilise les techniques les plus récentes d'Intelligence Artificielle. Il repose sur un dialogue explicatif avec l'élève. Pour atteindre cet objectif, nous avons dû construire une organisation des connaissances originale sur trois aspects :

La formalisation des métaconnaissances.

Les heuristiques de nature stratégique (comment orienter la stratégie de résolution ?, dans quel ordre examiner les règles pour parvenir à la méthode la plus économique ? ...) sont explicitées et justifiées.

L'expression des connaissances à différents niveaux de décomposition.

Les étapes de la résolution des diverses tâches (factoriser, résoudre une équation...) peuvent être plus ou moins détaillées par le système, chaque niveau de décomposition renvoyant à des concepts mathématiques distincts.

La formalisation du réseau sémantique

Les connaissances conceptuelles sont reliées aux savoir-faire procéduraux et stratégiques.

La modélisation de la connaissance experte dans le système NAIADE (généralisable à divers domaines de connaissances) contient donc en elle-même la possibilité d'explications diversifiées et adaptées au profil cognitif des utilisateurs.

NAIADE, l'environnement d'apprentissage pour le calcul algébrique que nous construisons à l'heure actuelle, simulera un enseignement du type travaux dirigés pour des élèves de collège (agés de douze à seize ans). La résolution des équations du premier degré ou qui s'y ramènent par factorisation, les méthodes heuristiques de factorisation des polynômes de degré supérieur ou égal à deux,.... constituent notre domaine test pour valider notre approche.

NAIADE est un projet pluri-disciplinaire qui implique des informaticiens, des psychologues et des enseignants de Mathématiques. La base de connaissances du système a été conçue à partir de recherches expérimentales (recueil de protocoles ponctuels pour inventorier et interpréter les erreurs des élèves sur un sujet donné et d'un corpus d'entretiens entre un élève et un enseignant sur plusieurs semaines consécutives pour apprécier la progression de l'élève).

Les principes de constitution de la base de connaissances du module expert ont été définis à partir d'une théorie du fonctionnement cognitif de l'élève en résolution de problèmes (cf Anderson 1987, Brown 1987, Chi et al 1982) et d'une analyse des processus de construction des connaissances (cf Cauzinille-Marmèche et al 1988).

Actuellement, une maquette de NAIADE intègre le module expert et l'analyseur du module de dialogue en langage naturel sur les quatre modules prévus : le module expert, le module de diagnostic et de modélisation cognitive de l'élève, le module pédagogique, le module de dialogue en langage naturel. Le système est mis en œuvre sur une station de travail HP 9000, série 350 et est écrit en langage LISP.

L'objet de cette communication est de présenter les schémas d'interventions explicatives que le système utilisera pour structurer le dialogue avec l'élève et en assurer la cohérence. Le contenu de ces schémas est bien sûr conditionné par les fonctionnalités du module expert qui résout le problème posé à l'élève.

L'ORGANISATION DE LA BASE DE CONNAISSANCES EXPERTE

La base de connaissances experte est composée de procédures, de règles et de métarègles. Les procédures et les règles modélisent des savoir-faire opératoires tandis que les métarègles modélisent des savoir-faire stratégiques qui orientent la résolution ou opèrent un choix parmi différentes méthodes applicables à une étape donnée du raisonnement. Ces connaissances sont structurées autour d'objectifs bien définis tels que "développer", "factoriser", "résoudre",....Plusieurs fonctionnalités sont visées au travers de l'organisation des connaissances expertes : dégager des critères pertinents de sélection des connaissances, expliciter et justifier la métaconnaissance, adapter le grain de la résolution à l'utilisateur, lier les savoir-faire aux concepts du domaine sous-jacents (cf Joab et al 1989)

1. LA SELECTION DES CONNAISSANCES

La sélection des connaissances s'opère grâce à un filtre de descripteurs hiérarchisés d'une expression (cf Joab 88) pour une tâche donnée. Les tâches sont ainsi analysées en fonction de la morphologie des expressions.

Par ailleurs, cet arbre des descripteurs explicite certains liens entre les méthodes de résolution. Ainsi, par leur niveau d'insertion dans la structure même de l'arbre, des connaissances apparaissent comme plus spécifiques que d'autres.

2. LA METACONNAISSANCE

Nous avons distingué quatre types de métarègles. Les métarègles de réduction indiquent dans quels contextes réduire les expressions. Les métarègles d'orientation orientent la résolution par l'identification d'une classe de problèmes. Elles interviennent à une étape clé du raisonnement. Certaines d'entre elles inscrivent une tâche prioritaire dans l'agenda des tâches.

L'exécution de cette tâche prioritaire permet la reconnaissance de la classe de problèmes pertinente. Les métarègles de simplification ramènent le problème posé à un problème plus simple. Les métarègles d'économie sélectionnent la méthode de résolution la plus économique.

Cette typologie fait apparaître explicitement la fonction globale des métarègles (réduire, simplifier, ...). Le système expert active dans l'ordre les métarègles de réduction, les métarègles d'orientation, les métarègles de simplification et enfin les métarègles d'économie.

3. LES NIVEAUX DE DECOMPOSITION

Le module expert présente à l'élève un modèle de la résolution du problème. Pour s'adapter à l'utilisateur, le système peut, au besoin, détailler ou au contraire synthétiser les étapes intermédiaires. Il s'agit là de "compiler" ou de "décompiler" une connaissance opératoire en utilisant dans la résolution, soit une procédure, soit un paquet de règles associées à une tâche. La construction de la base de connaissances fait apparaître des niveaux de connaissances définis à partir de couches homogènes des concepts du domaine. Par exemple, le niveau conceptuel le plus élémentaire pour notre domaine, l'algèbre, fait référence aux calculs dans le corps des nombres réels, le niveau supérieur fait intervenir la notion de calcul littéral, les notions de monôme,de variable, de degré et de coefficient numérique d'un monôme, ...

4. LE ROLE DU RESEAU CONCEPTUEL

Le réseau sémantique illustre l'organisation des concepts du domaine. Les concepts sont liés aux règles et aux métarègles expertes. L'intérêt de cette organisation des connaissances est essentiellement de pouvoir mettre en relation les savoir-faire procéduraux et stratégiques et les connaissances conceptuelles. Ainsi, le système pourra présenter un raisonnement sous différents angles : procédural, stratégique ou conceptuel.

Les différentes connaissances mises en œuvre par le système expert pourront intervenir dans le dialogue explicatif. Nous structurerons les interventions explicatives du système pour présenter une résolution à l'élève grâce aux schémas d'explication.

Les connaissances nécessaires pour résoudre le problème ne sont pas toujours suffisantes pour expliquer, commenter et justifier la solution (cf Clancey 1983, Nicaud 1987, Kassel 1988, Vivet 87). D'autres connaissances, qui justifient et analysent l'expertise, davantage qu'elles n'opèrent sur la résolution, entrent alors en jeu et sont articulées dans les schémas d'explication.

LES SCHEMAS D'EXPLICATION

On peut distinguer deux types de présentation du raisonnement : une vision locale pour expliciter et justifier pas à pas chaque étape de la résolution et une vision plus synthétique qui ne mettra en avant que les étapes clé de la résolution pour favoriser l'acquisition du mécanisme de planification.

1. LA PRESENTATION D'UNE ETAPE

Les schémas d'explication exposent et structurent la succession des interventions possibles du système, pour la justification d'une étape de résolution. Ces schémas ont pour rôle d'assurer la cohérence du dialogue explicatif. Les différentes interventions potentielles du système sont partiellement ordonnées. Nombre d'entre elles resteront implicites dans l'interaction réelle avec l'utilisateur, il serait en effet fastidieux pour lui d'obtenir systématiquement toutes les informations contenues dans les schémas.

Lorsque l'intervention porte sur un concept, celui-ci sera toujours exposé dans le contexte de la résolution en cours pour en favoriser l'appropriation par l'élève.

Le premier schéma d'explication caractérise l'état courant du problème. Il a pour objet de situer le but actuel dans le plan courant, d'en mesurer l'écart avec l'objectif plus général en cours de traitement.

A_1 citer le but courant B
A_2 situer B dans le plan courant P
$A_{3.1}$ présenter l'objectif général lié à B : obtenir une forme F
$A_{3.2}$ mesurer l'écart entre B et F
$A_{3.3}$ indiquer les concepts sous-jacents

fig. 1 : SCHEMA 1 "ETAT COURANT DU PROBLEME"

L'exemple suivant illustre le premier schéma. Il est extrait de la résolution de l'équation :

$$(3x - 1)^2 - (x + 4)^2 = 0$$

A_1 Je factorise $(3x - 1)^2 - (x + 4)^2$
A_2 Je factorise $(3x - 1)^2 - (x + 4)^2$ pour résoudre $(3x - 1)^2 - (x + 4)^2 = 0$
$A_{3.1}$ Je veux obtenir un produit de facteurs dont chacun est un polynôme ordonné et réduit de degré 1 ou somme de deux carrés[1].
$A_{3.2}$ Or $(3x - 1)^2 - (x + 4)^2$ est la différence des deux termes $(3x - 1)^2$ et $(x + 4)^2$
$A_{3.3}$ Un polynôme ordonné et réduit est un polynôme réduit et ordonné suivant les puissances décroissantes de la variable; par exemple, x + 4 est un polynôme ordonné et réduit,...
fig. 2 : EXEMPLE 1

Le deuxième schéma justifie la métaconnaissance activée par le système expert pour résoudre le problème courant. Certaines interventions de ce schéma traduisent le contenu opératoire de la métarègle alors que d'autres interventions justifient la stratégie, soit de façon générale, soit de façon détaillée.

B_1 indiquer les conditions de déclenchement de la métarègle
B_2 indiquer la sélection des connaissances opérée
B_3 indiquer la fonction (réduction, simplification ou économie)
B_4 appuyer la justification générale par une justification détaillée dans le contexte de l'exemple traité. $B_{4.1}$ réduction $B_{4.2}$ simplification $B_{4.3}$ économie indiquer l'intérêt de la méthode sélectionnée
fig. 3 : SCHEMA 2 "CONNAISSANCES STRATEGIQUES"

L'exemple suivant illustre le deuxième schéma pour une métarègle d'économie.Le but courant est alors "développer $2x(x+1)3x^2$"
La métarègle déclenchée indique que pour un produit de plus de deux facteurs, il est préférable de regrouper les monômes et d'en faire le produit avant d'associer les autres facteurs.

[1] Ceci, dans le champ de connaissances d'un élève de Collège.

B_1 On effectue un produit de 3 facteurs
B_2 Regrouper d'abord les facteurs numériques, puis les monômes
B_3 Cette démarche est plus économique
$B_{4.3}$ En effet , dans le développement, il vaut mieux effectuer d'abord les produits élémentaires avant de développer les facteurs qui sont des sommes.
fig. 4 EXEMPLE 2

Le troisième schéma présente l'exécution de la règle déclenchée à cette étape de la résolution. Cette règle a été sélectionnée, soit par une métarègle, soit uniquement par le filtre des descripteurs de l'expression.

C_1 indiquer les conditions de déclenchement de la règle
C_2 Etudier les résultats du déclenchement, en particulier leur influence sur le plan courant
C_3 Indiquer en quoi on se rapproche de la forme attendue
C_4 Justifier conceptuellement la règle
fig. 5 : SCHEMA 3 "CONNAISSANCES PROCEDURALES"

Dans l'exemple 3, le but courant est :
"développer $2x(x+1)3x^2$", la règle appliquée associe les facteurs du produit qui sont des monômes.

C_1 $2x(x+1)3x^2$ est un produit de 3 facteurs, 2x et $3x^2$ sont des monômes, (x+1) est le facteur restant du produit.
C_2 Pour développer $2x(x+1)3x^2$, il faut d'abord effectuer $(2x) \times (3x^2)$ et associer le résultat au facteur (x + 1).
C_3 Le nombre de facteurs du produit après ce calcul, sera égal à 2.
C_4 Le produit dans $\mathbb{R}$ est associatif et commutatif.
fig. 6 : EXEMPLE 3

Le système ne peut présenter uniquement la résolution d'un problème, étape par étape. L'utilisation des schémas précédents conduit plutôt à ajouter des informations supplémentaires aux connaissances opératoires mobilisées dans le processus de résolution.

Les schémas explicatifs ne parviennent pas à rendre compte du mécanisme global de la résolution, en particulier lorsque les métarègles d'orientation introduisent un plan d'actions ou une tâche prioritaire pour déterminer la classe de problèmes à laquelle se référer et appliquer la méthode de résolution adaptée. Il est nécessaire d'intégrer cette analyse de la tâche vis à vis des classes de problèmes dans une vision synthétique du raisonnement.

2. LA SYNTHESE DU RAISONNEMENT

Une explication du raisonnement doit mettre en évidence un plan global. On intègrera dans cette synthèse, la présentation des métarègles d'orientation ainsi que celle des étapes clé de la résolution.

Le schéma 4 présente les métarègles d'orientation. Il justifie l'introduction d'un plan ou d'une tâche prioritaire par les classes de problèmes sous-jacentes.

D_1 indiquer les conditions de déclenchement de la métarègle
D_2 indiquer la tâche ou le plan prioritaire
D_3 indiquer la fonction d'orientation
D_4 indiquer les classes de problèmes.

fig. 7 : SCHEMA 4 "ORIENTATION"

L'exemple suivant illustre le schéma 4. Il est extrait de la résolution de l'équation :
$(3x - 1)^2 - (x + 4)^2 = 0$
La métarègle considérée introduit la recherche du degré de l'équation comme tâche prioritaire.

D_1 L'équation $(3x - 1)^2 - (x + 4)^2 = 0$ n'est pas de la forme "produit de facteurs nul"
D_2 Il faut rechercher le degré de l'équation
D_3 Le degré permettra de déterminer la méthode de résolution.
D_4 On distingue les équations de degré supérieur ou égal à 2, des équations de degré 1

fig. 8 : EXEMPLE 4

Pour récapituler les étapes clé de la résolution, il nous faut privilégier les tâches de plus haut niveau. Les résultats atteints sont mémorisés pour synthétiser la résolution du problème.

PERSPECTIVES

Pour atteindre notre objectif de dialogue explicatif, il convient maintenant d'expliciter les conditions d'activation des interventions pour les différents schémas définis. Ces conditions pourront porter sur des éléments du profil de l'utilisateur. Il faut tenir compte également des requêtes explicites de l'utilisateur en déclenchant un mécanisme de planification du dialogue en situant la requête par rapport au schéma actif. Ces principes visent à assurer une certaine économie des interventions du système, sa coopérativité et la cohérence structurelle de l'explication.

BIBLIOGRAPHIE

Anderson J.R. (1987). *Skill acquisition : compilation of weak-method problem solutions*. Psychological Review, 1987, 94, p 192-210

Brown A. (1987). *Metacognition , executive control, self-regulation, and other more mysterious mechanisms* . in F.E. Weinert et R.H. Kluwe (Eds.). Metacognition, motivation and understanding, Lawrence Erlbaum, Hillsdale.

Cauzinille-Marmèche E., Joab M., Mathieu J., Monteils N. (1988).Explicitation et représentation des connaissances des élèves de Collège en Algèbre. *Didactique et Acquisition des connaissances scientifiques* . G. Vergnaud, G. Brousseau, M. Hulin et GRECO Didactique C.N.R.S.(Eds.), La Pensée Sauvage, Paris.

Chi M.T.H., Glaser R., Rees E. (1982) . *Expertise in problem solving*. in R. Sternberg (ed.), Advances in the psychology of human intelligence, vol 1, Laurence Erlbaum Ass., Hillsdale NJ, pp 7-75.

Clancey W. J., (1983). *The epistemology of a rule-based expert system - a framework for explanation*. Artificial Intelligence, vol 20, p 215-251.

Joab M.(1988) . Le système NAIADE : modélisation des connaissances et de leur utilisation dans la résolution de problèmes. *Actes du congrès I.T.S. 88*, Montréal, 1-3 juin 1988.

Joab M., Cauzinille-Marmèche E., Mathieu J (1989) . NAIADE, un
système d'aide au calcul algébrique. *Actes de la Convention IA* , 23-27 janvier 1989, Paris.

Kassel G. (1988). *Deux directions de recherche pour l'explication du raisonnement*. Actes du colloque "Les systèmes experts et leurs applications" , Avignon 1988.

Nicaud, J.F. (1987). *APLUSIX : un système expert de résolution pédagogique d'exercices d'algèbre*, thèse de l'université Paris-Sud, décembre 1987, Orsay.

Vivet M. (1987) . *Systèmes experts pour enseigner : méta-connaissances et explications*, Actes du colloque COGNITIVA 87, Paris.

PRECEPTOR: AN ALGEBRAIC WORD PROBLEM TUTOR

K. Gladwell Schulze
Computer and Information Science Department
Southeastern Massachusetts University
North Dartmouth, MA, USA

Studies have shown that only a limited amount of information can be maintained in a student's working memory at a given time. PRECEPTOR is an open-ended, hybrid, intelligent tutoring system that increases a student's success rate at solving an algebraic word problem involving two variables by improving his ability to map the natural language representation into an equivalent abstract algebraic equation. The system dynamically generates word problems which are partitioned into three classes according to the amount of requisite information that is explicitly provided by the word problem. Each problem classification is further subdivided to provide problems at four levels of difficulty for which the system provides both individualized and differential tutoring. The system currently contains a tutoring strategy for each level of difficulty, two tutoring strategies or plans for each problem classification and adapts to the individual student by determining the most effective strategy for him. As the student progresses through the four levels of difficulty for each classification, an increased demand is placed on his working memory, thereby improving his conceptualization and generalization skills.

PRECEPTOR : LOGICIEL D'AIDE À LA SOLUTION DE PROBLÈMES ALGÉBRIQUES EXPRIMÉS EN MOTS

D'après certaines études, le mémoire de travail d'un étudiant ne peut emmagasiner qu'un nombre limité d'informations à un moment donné. PRECEPTOR est un système d'initiation intelligent, hybride, évolutif qui augmente le taux de réussite d'un étudiant dans la solution d'un problème d'algèbre exprimé en mots et comportant deux variables en améliorant sa capacité à transformer la représentation en langue naturelle d'un problème en une équation algébrique abstraite et équivalente. Le système génère de façon dynamique les problèmes exprimés en mots lesquels sont subdivisés en trois groupes selon le nombre d'informations requises explicitement fournies dans le problème. Chaque catégorie de problèmes est ensuite subdivisée en quatre niveaux de difficulté pour lesquels le système assure une aide individualisée et particulière. Le système comporte une stratégie d'initiation pour chaque niveau de difficulté, deux stratégies ou plans d'initiation pour chaque catégorie de problème et s'adapte à l'étudiant en déterminant la stratégie la plus efficace pour lui. Au fur et à mesure que l'étudiant progresse à travers les quatre niveaux de difficulté de chaque catégorie, le système sollicite davantage sa mémoire de travail, améliorant ainsi ses aptitudes à conceptualiser et à généraliser.

PRECEPTOR: AN ALGEBRAIC WORD PROBLEM TUTOR

Kay Gladwell Schulze

Southeastern Massachusetts University
North Dartmouth, Massachusetts, USA

It is an accepted fact that many students perform extremely well within the framework of formal mathematics and yet experience great difficulty in making a transformation from this formal knowledge to real world experience. This is particularly true for an algebra student who is faced with the problem of solving word problems involving two variables. The goal of this research has been to develop an intelligent tutoring system that will increase a student's success rate at solving algebraic word problems involving two variables by improving his ability to map the natural language representation into an abstract algebraic equation.

PRECEPTOR

PRECEPTOR[1] (SCH 88) is an intelligent tutoring system that tutors students in the transformation of three different classifications of algebraic word problems into equivalent abstract algebraic equations. It was developed for a student audience which exhibits competency in the computational aspects of algebra but experiences difficulty with word problems. One of its goals was to help the student determine the explicit information contained in the problem and the implicit information that needed to be transferred to his working memory in order to solve the problem.

All three problem classifications used by PRECEPTOR involve equations with two unknown variables. The first, known as **money** problems, deals with the purchase of two similar types of merchandise at different prices. All problems in this class explicitly contain all the information necessary to generate the word problem. A level one example is:

> Train fares from Boston to Philadelphia cost $97 for coach seats and $115 for first class seats. If the receipts for 113 people were $12833, how many were first class seats?

[1]PRECEPTOR currently runs on a Macintosh II with two megabytes of memory.

The second classification, referred to as **coin** problems, involves a collection of two types of coins. These problems differ from **money** problems because the value of each coin is only implicitly available from the problem but like the **money** problems is a required quantity for correct equation generation. This classification was included to determine if the student can make the transition from a simpler, explicitly stated word problem to a similar, implicitly stated one. A level two coin problem is:

> A piggy bank contains $1.75 consisting of nickels and dimes. If the number of dimes is twice the number of nickels, how many are nickels?

The third classification of problems involves the **perimeter** of three geometric figures - rectangles, parallelograms and isosceles triangles. In order to generate a correct equation an appropriate perimeter formula must be used. Even though the existence of such formulas will probably reside in the student's memory, the actual formulas themselves may have to be self generated or provided by tutor. An example of a level four problem is:

> The perimeter of a rectangle which is 4 times as long as it is wide, is 140 yards. How long is the width?

THE ARCHITECTUAL SCHEMA

A hybrid knowledge representation scheme consisting of production rules and Lisp functions was used in the implementation. The architectural scheme currently consists of four modules: the Generator, the Administrator, the Matcher, and the Tutor. The Generator dynamically generates the word problems, their equations and solutions while the Administrator controls the session with the student by invoking the tutor and generator modules. The Matcher matches the equations generated by PRECEPTOR with those generated by the student and the Tutor controls the tutoring and the equation matching processes.

Although only the generation of a **final** equation is required of the student, the process of obtaining this equation involves two other equations. The first, called the **primary** equation, involves either the total monetary value of the coins, the total purchase price or profit, or the perimeter of a geometric figure. The format for this equation in all three classifications is:

$AX + BY = C$ where A, B and C are constants.

The second equation, called the **secondary** equation, involves the relationship between the number of coins, the number of pieces of purchased merchandise or the relationship between the length of the sides of the geometric figure.

The problems are dynamically generated from templates based on problems in current algebra texts (PRO 86) and were designed with four different difficulty levels based on the **secondary** equation.

THE GENERATOR

When PRECEPTOR generates a word problem, it provides the Generator with a classification and difficulty level. The Generator then randomly generates a template and the necessary fillers for it. The **primary** and **secondary** equations are then calculated.

In all three classifications of word problems the general format of the **primary** equation is

$$AX + BY = C.$$

The difference in the classes is determined by whether or not the format of the equation resides in the student's long term memory and whether the values for A and B are explicitly or implicitly contained in the word problem.

A level one **secondary** equation has the form:

$Y = X \pm K$ where K is a constant

The second level has the form:

$Y = KX$ where K is again a constant.

The third level of difficulty combines levels one and two and has the form:

$AY \pm B = CX \pm D$ where A, B, C, D are constants.

The final level involves problems that are simply an extension of the first or second level. Although they have the same form as these levels, the problems are **student-professor** problems (CLE 82).

Once the **primary** and **secondary** equations are determined, they are solved together to obtain the **final** equation which is returned to the Administrator.

THE ADMINISTRATOR

Except for communication with the student during the tutoring phase, the Administrator provides all of the interface communication between PRECEPTOR and the student, controls the classification and level of the word problems and maintains the student's records.

The problem solving cycle of PRECEPTOR is based on that which was used in the system reported by Sims-Knight and Kaput (SIM 87). The student enters the cycle at a particular level for a specific classification of problems and exits by *topping out* or *bottoming out*. He *tops out* when he successfully generates the final equation for four problems of the same level. At this point, he is given the choice of exiting or continuing to the next level.

A student can *bottom out* at several points in the cycle. A record is maintained of the number of problems he has missed and he may exit the system after he misses his tenth, fifteenth or twentieth problem. Multiple exit plateaus were provided based on the results of the research of Sims-Knight (SIM 87).

When a student generates the correct equation to a problem, regardless of whether or not tutoring was involved, he is asked to solve his equation for the numeric solution of the primary variable. However, no attempt is made to tutor the student in this process.

THE MATCHER

The Matcher is first invoked by the Generator in an attempt to match the student's final equation with PRECEPTOR's. The Matcher converts the student's equation into one containing only *X* and/or *Y* and then determines whether the student used explicit multiplication or juxtapositioning, adjusting PRECEPTOR's equation accordingly, before doing any comparisons.

The system always generates a final equation that involves the primary variable (i.e. the one that represents the request of the word problem). If the student's equation does not match PRECEPTOR's final equation, the system explores the possibility that he expressed the final equation in terms of the secondary variable.

If neither of the above two matches succeed, the Matcher will compare the student's equation against PRECEPTOR's primary equation. If a match occurs, the student is asked to find

another equation involving two variables. Upon successfully generating a correct secondary equation, he is asked to combine the two equations into a single final equation. However, should the student either be unable to generate a secondary equation or generates the wrong one, he is tutored for the secondary equation based on the level of difficulty of the problem. Should the student's equation fail to match either PRECEPTOR's final or primary equation, it is compared to PRECEPTOR's secondary equation. If no match occurs, the student is tutored in how to generate the primary equation given the specific problem classification.

THE TUTOR

While several investigators (REI 85; JOH 85) have attempted to model the student's cognitive processes, the present research models the cognitive processes of an effective teacher by providing differential and individualized tutoring based on the student's tutorial history and the problem's difficulty level.

If a teacher is to be effective, he must be capable of explaining a concept in multiple ways. Regardless of the effectiveness of the "traditional" strategy, there is always some student who still will not understand. For him, the teacher must deviate to another strategy. In addition, a good teacher will remember to bypass the "traditional" approach and use the "back-up" approach for any future tutoring with that student, until or unless the second approach also begins to fail. At this point, the teacher will retrieve a third strategy from his reservoir of tutoring strategies.

The current version of PRECEPTOR provides two strategies for tutoring the primary equation for each of the problem classifications. The first strategy for both the money and coin classifications uses the strategy that the whole is equal to the sum of its parts. Therefore, it helps the student determine the value for M of item X and N of item Y and then attempts to get him to combine these into the value of $MX+NY$. The second strategy helps him determine the value of $MX+NY$ for various values of M and N without separating it into parts. The first strategy for the perimeter classification attempts to have the student recall from his working memory the perimeter formula, while the second strategy forces the student to derive the formula from the definition of perimeter.

It was never the intention of PRECEPTOR to tutor students in how to solve two simultaneous equations and hence little tutoring is given if the student has generated (with or without help) the primary and secondary equations and cannot generate the final equation.

Tutoring for a primary equation only occurs if the student fails to generate any equation when asked to give the final equation or if he generates an equation and it fails to match PRECEPTOR's final, primary or secondary equations. Before extensive tutoring occurs, the system first determines if the student has generated the correct left-hand side (LHS) of the primary equation. If this situation exists, PRECEPTOR simply tutors him in obtaining the correct right hand side. However, if the LHS is wrong, the system, after having determined which strategy is most effective for him, provides appropriate tutoring. The system is biased towards strategy one if the student is equally successful with each.

PRECEPTOR uses its knowledge of the difficulty level and the inherent properties and format of each level in tutoring the student in the generation of the secondary equation. Since the secondary equation does not differ in format with the respective classifications, the tutoring is independent of classification. Although there are two distinct tutoring strategies for a primary equation, the current system contains only one strategy for each difficulty level.

The student may enter the tutoring phase for the secondary equation only after he has generated the correct primary equation and because he was either totally incapable of generating a secondary equation or because he generated one which failed to match PRECEPTOR's. The system is aware of the state in which the student entered the tutoring phase and provides appropriate tutoring accordingly.

Level one secondary equations can assume one of three distinct forms, all of which are a variation of the format

$$Y = \pm X \pm C \qquad \text{where C is a constant.}$$

The system first determines the appropriate form of the equation. If the student needs tutoring for an equation involving the concept of equality, PRECEPTOR first determines if the student is aware of the relationship that exists between the two variables. If necessary, the concept of equality is explained to the student and

he is then asked to express this concept in terms of an algebraic equation involving two variables.

When the secondary equation is of the form X+Y=K, PRECEPTOR determines if the student understands the concept of summation and merely set the sum equal to the wrong value. When this occurs, he is requested to correct his equation and then reenters the problem solving cycle. Otherwise he is given complete tutoring on the concept of summation.

If tutoring is needed for a secondary equation of the form $X = Y \pm K$, it may be necessary for PRECEPTOR to request additional information from the student before it can determine if his equation is correct. For example, if the equation generated by PRECEPTOR is

$$Y = X + 9$$

and the student generates the equation

$$B - C = 9 \quad \text{or} \quad B - C - 9 = 0$$

the system must determine whether Y represents the same thing as B or as C. If Y and B represent the same thing, the student returns to the problem solving cycle. However, if it does not the student continues with the tutoring for this form of equation.

Secondary equations of level two are of the form $X = AY$ where A is a constant. Because A can be either a whole number or a fraction, it is not possible for the system to immediately determine if the student's equation matches PRECEPTOR's. For example the system's equation may be X=(2/5)Y and the student may have generated 5X=2Y. When this situation occurs PRECEPTOR verifies that both X's represent the same thing and the student proceeds accordingly. When tutoring is necessary, the student is helped to determine and express the relationship that exists between the two variables.

Since secondary equations of level three combine the concepts of levels one and two, the student is first tutored to conceptualize the level two portion and then the level one. After each conceptualization, he is requested to express the concept as an equation.

Even though problems of level four are of the **student-professor** classification, their format is the same as level one or level two. A previous study (SCH 87) indicates that students have fewer problems generating the equation if the format is of level one. PRECEPTOR first determines the correct format and then has the student indicate which variable represents the larger value and the operation necessary for the two values to equal each other. Finally, he is asked to express this concept as an equation.

After the primary and secondary equations have been obtained, PRECEPTOR determines whether the variables of the two equation are consistent. If an inconsistency exists, the student is forced to resolve it before a match is attempted on a final equation.

CONCLUSIONS

This research has shown that it is possible to dynamically generate algebraic word problems in two unknown variables and to model a teacher by providing individualized and differential tutoring. PRECEPTOR is currently being tested to determine its effectiveness.

REFERENCES

Clement, J. (1982) Algebra word problem solutions: Thought processes underlying a common misconception, *Journal for Research in Mathematics Education*, 13: 16-30.

Johnson, L. & Soloway, E. (1985) PROUST: knowledge-based program understanding, *IEEE Transactions on Software Engineering*, **SE-11** (3): 267-275.

Proga, R. (1986) Arithmetic and Algebra, Prindle, Webber & Schmidt:Boston, MA.

Reiser, B., Anderson, J. & Farrell, R. (1985) Dynamic student modeling in an intelligent tutor for LISP programming, *Proceedings of the Ninth IJCAI*, Los Angeles, CA, 8-14.

Schulze, K. (1987) Error generation in the translation of algebraic word problems into equivalent equations, (SMU-CIS Technical Report No. 87-001), Southeastern Massachusetts University, North Dartmouth, MA.

Schulze, K. (1988) PRECEPTOR: A tutor for transforming algebraic word problems into abstract algebraic equations, Ph.D. Dissertation, Boston University, Boston, MA.

Sims-Knight, J. (1987) Personal communication.

Sims-Knight, J. & Kaput, J. (1987) Progress Report, Year 2, NSF Grant No. MDR-8410316.

EMBEDDED TRAINING AND INTELLIGENT USER INTERFACES FOR APPLICATION SOFTWARE

R.J. Clark
Intelligent Workbench Corporation
Cumberland, ON

The 30 million microcomputers sitting on corporate desks today are being grossly underutilized from the perspective of personal productivity improvement. Personal computing is a skills-intensive activity and organizations have been unable to provide their information workers with the on-going knowledge and skills they require to make effective use of the available hardware and software tools. More and better short courses are not the solution; skills acquired, and not immediately and continuously applied, are lost in a few weeks.

We need, instead, to embed the skills transfer capability into the software. Functional training must be provided as brief user-initiated on-line tutorials, detailed in-context help and intelligent user interfaces which can understand and carry out a user's request to provide assistance in performing a function. As well, the application must have the capability to systematically diagnose user skills deficiencies and implement an optimum strategy for correcting them.

The author has spent six years designing software with such attributes. He will discuss the objectives, techniques and implications of intelligent dialogue and screen interface design, and will illustrate the presentation with sample dialogues from a number of application software packages.

FORMATION EN ARRIÈRE-PLAN ET INTERFACES UTILISATEURS INTELLIGENTES DANS LES LOGICIELS D'APPLICATION

Les 30 millions de micro-ordinateurs qu'on retrouve aujourd'hui dans les bureaux des compagnies sont manifestement sous-utilisés quand on considère leur utilisation du point de vue de l'amélioration de la productivité personnelle. L'informatique personnelle demande beaucoup de compétences, et malheureusement les organisations ont fait la preuve de leur incapacité à donner une formation continue à leurs employés d'information pour que ces derniers aient les connaissances et les compétences leur permettant d'exploiter le mieux possible les outils matériels et logiciels disponibles. Et la solution ne consiste pas à augmenter le nombre de petits cours qu'on leur fait suivre ni à en améliorer la qualité, car on sait bien que les compétences acquises qui ne sont pas appliquées immédiatement et de façon continue se perdent dans les semaines qui suivent.

La solution que nous proposons est d'avoir en arrière-plan dans le logiciel une fonction de transfert de connaissances. L'enseignement des fonctions peut se faire au moyen de brefs tutoriels en ligne lancés par l'utilisateur, d'aide en contexte détaillé et d'interfaces intelligentes qui peuvent comprendre la demande d'un utilisateur qui veut de l'aide pour exécuter une fonction et y répondre. En plus, l'application doit pouvoir diagnostiquer systématiquement les carences de l'utilisateur et mettre en oeuvre la meilleure stratégie pour les corriger.

L'auteur a consacré six années à la conception de logiciels dotés de telles caractéristiques. Il analysera les objectifs, les techniques et les conséquences d'un dialogue intelligent et la conception d'interfaces d'écran, puis il complètera son exposé en donnant des exemples de dialogue dans le cas d'un certain nombre de logiciels d'application.

EMBEDDED TRAINING AND INTELLIGENT USER INTERFACES FOR APPLICATION SOFTWARE

R.J. Clark
Intelligent Workbench Corporation
Cumberland, ON

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

THE USEFULNESS OF EXPERT SYSTEMS AS INTELLIGENT ON-THE-JOB TRAINING AIDS

K.L. Dalkir
Canadian Workplace Automation Research Center
Department of Communications
Laval, PQ

The traditional uses of computers in education have included testing, drills, tutorials, games, simulations and student management systems. The advent of advanced technologies such as artificial intelligence and interactive videodisks open the way to new computer applications in education and training. One of particular interest involves the integration of expert systems with computer-assisted instruction --"embedded knowledge-based training systems".

The rate of technological advances and high employee turnover create a strong demand for new tools that are self-explanatory and therefore easier to use. Office tools with embedded instructions help facilities and useful references will not only reduce learning times but also do away with a great deal of frustration and costly errors. Expert systems appear to be good candidates for such embedded training and consulting systems. A general knowledge engineering methodology is proposed to efficiently add on training modules to existing knowledge base systems. This type of hybrid intelligent system will provide more sophisticated assistance to both novice and experienced users.

L'UTILITÉ DES SYSTÈMES EXPERTS COMME OUTILS INTELLIGENTS DE FORMATION EN COURS D'EMPLOI

L'utilisation traditionnelle des ordinateurs dans le domaine de l'enseignement comprenait les examens, les exercices, les guides d'initiation, les jeux, les simulations et les systèmes de gestion des étudiants. L'avènement des technologies avancées, comme l'intelligence artificielle et le vidéodisque interactif, ouvre la voie aux nouvelles applications informatiques dans le domaine de l'enseignement et de la formation. L'une des applications les plus intéressantes met en cause l'intégration de systèmes experts et l'enseignement assisté par ordinateur, les "systèmes experts de formation intégrés".

La vitesse des progrès technologiques et le haut taux de roulement du personnel entraînent une forte demande de nouveaux outils explicites et, par conséquent, plus faciles à utiliser. Les outils de bureau avec instructions intégrées, les fonctions d'aide et les références utiles ne dimineront pas seulement les durées d'apprentissage, mais élimineront également beaucoup de frustration et d'erreurs coûteuses. Les systèmes experts semblent être de bons candidats pour ces systèmes intégrés de formation et de consultation. On propose une méthodologie de cognitique générale pour ajouter efficacement des modules de formation aux systèmes experts existants. Ce type de système intelligent hybride offrira des fonctions d'aide plus perfectionnées aux utilisateurs novices et expérimentés.

THE USEFULNESS OF EXPERT SYSTEMS AS INTELLIGENT ON-THE-JOB TRAINING AIDS

K. L. Dalkir

Expert Systems Group, Division of Advanced Technology

Canadian Workplace Automation Research Centre, Communications Canada

The general issue of using expert systems as intelligent job aids is addressed in this paper. Such knowledge-based computer assistants can transform expert systems into useful consultants and trainers. On-the-job training applications can be developed by modifying an existing expert system. A general knowledge engineering methodology is proposed in order to efficiently 'add on' a training module to existing knowledge based systems. Organizational factors in the implementation of such systems are considered. Finally, our own efforts to address training needs expressed by expert system users are discussed.

INTRODUCTION

Demographic projections point to a major shortage of trained, educated people[1]. We are increasing the skills of the computer but not those of computer users. The unfortunate scenario that can result is that of very competent automated systems being operated by only marginally skilled users. The answer is not always formalized training to upgrade skills. The alternative is to use artificial intelligence techniques, especially expert systems, in order to produce both competent machines and competent workers.

The approach discussed here is to develop embedded knowledge bases systems that are capable of training, overseeing and supporting their users. At one end of the training spectrum are conventional approaches such as training courses and manuals. At the other extreme are highly sophisticated intelligent tutorial systems which contain not only subject matter knowledge (domain expertise) but also guidelines on how best to convey this subject matter to the student (pedagogical expertise). The use of a rule-based expert system as an intelligent on-line assistant falls roughly in the middle of this range. The major distinction relates to the degree of sophistication of the explanations given to users. Expert systems with enhanced explanation facilities can be used as job aids to increase the competence of users.

THE EXPERT SYSTEM AS JOB AID

A job aid is a set of easy to use reference materials that often substitutes for formal training and manuals[2]. The job aid focuses on a specific problem and consists of checklists, forms, tables, decision trees and so forth to help "walk" the user through the job at hand. The more complex a job, the more likely it is that a job aid will be needed instead of, but more often in addition to, training. For example, if the content is not easily memorized, some type of quick reference card would be useful to have on hand such as those used for word processing applications.

In fact, the same dimensions are assessed when identifying appropriate expert system applications. Good expert system candidates are tasks that are infrequently performed, complex, consist of a large number of steps and have serious error consequences. This list can be shortened to the following question: "does the task require the use of a job aid (or should it)?" An expert system is really a very sophisticated form of job aid - an electronic on-line computer job aid.

The premise behind artificial intelligence approaches to training is to provide every student the opportunity to become apprenticed to a "master" in the form of an expert on a diskette[3]. Intelligent help facilities can provide more effective instruction since they are coupled with actual performance on the job (embedded system). Ideally, the training would be developed as the product (i.e. expert system) is being developed in order to maximize the accuracy of the advice to be given to users[4]. The advantage is that users are placed in a simulation situation where they are asked to perform the tasks they would on the actual job.

ORGANIZATIONAL CONSIDERATIONS

The improvement of certain features of expert systems, in particular helpful comments and explanations to users, can render the expert system a learning tool as well as a decision aid. In order for expert systems to become useful organizational tools, however, they must be fully integrated within the work environment. In practical terms, this often means the ability to link up the expert system to existing corporate databases and communication networks. Standalones often turn out to be truly alone in that the motivation to leave your workstation for an hour or so is usually not overwhelming. The expert system tool must be as easy to use and as ubiquitous as the telephone is on an office worker's desk[5]. In real time work environments, this aspect rapidly becomes the overriding factor to consider. To this end, expert system tools, regardless of their intended function, must be integrated within the existing organizational systems.

The intelligent job aid should also have a life cycle as long as that of the organization it is a part of. An organization can be said to be a viable system, in other words, they are in the business of surviving in the long run. The term "viable system" as used in this context refers to a system that adapts to changes in the real world while still maintaining its identity[6]. In the same way, training should be a viable endeavor as opposed to a one-shot deal[7]. Training, regardless of it form, is an ongoing concern and the need for training will never go away. In fact, as technology becomes more and more complex as well as more commonplace, training is going to have a hard time coping with the demand. Electronic or high technology training seems a good way to address this need[8].

PROPOSED METHODOLOGY - KNOWLEDGE ENGINEERING

The objectives of an intelligent help facility are : to have people use the expert system, to have them ask for any additional information they may need, and to have them make good decisions (comparable to those made by experts). Repeated use of expert systems should result in incidental learning of this additional explanatory information and, ideally, this learning will be transferred to the job to improve performance and productivity.

Expert system prototypes can be built quickly and cheaply and once the knowledge base is developed, you're already half way there to providing intelligent assistance to users of the system. There is the added advantage of portability since employees can use the job aid at their desks, at home whenever they want to. They can also use the expert system in relative privacy since any mistakes are not only unadvertised but have no real-life consequences. An expert system can thus offer all the benefits of a simulation plus the advantages of a knowledge base driving the simulation.

The general approach of protocol analysis, one of the techniques used in knowledge engineering, is a good means of task analysis to extract not only the rules of the domain but also the rules used by a counsellor or advisor for the domain. This approch was used by Aaronson and Carroll[9] to study the interactions between computer users and advisors at a "help desk" in the computer centre. By recording and analyzing the questions posed and answers offered, an intelligent help facility can be designed which mimics the human consultants. An expert system help facility can be developed based on such an empirical study of the advisory interactions.

Aaronson and Carroll found a number of intriguing results. For example, users tended to ask "is this correct?" rather than "what is the correct solution?" Thus the "answer is in the question" since the user is already providing a great deal of information on the context, any assumptions and the general approach used. The remainder of the questions were requests for more information, and a form of negotiating process to establish why the advisor's solution is better or preferred to that proposed by the user. Less sophisticated users tended to ask more verification type questions to "validate" or confirm their proposed solutions whereas more experienced users tended to try out new ideas about what was happening or what might be causing a problem.

The implication for a knowledge based helper is that not only are they a good way for users to have their ideas confirmed but also for them to remember the solutions, since they are assuming a fairly active role in the interaction. Successful interactions were found to be those in which there was a mixed initiative in the dialogue. The advisor only intervened when asked a direct question or when a misconception could be identified in the user's reasoning. There was a definite need to be able to render explicit the reasoning used by both the advisor and the user.

This is related to the type of learning conversation postulated by G. Pask[10] in his Conversation Theory. Two participants in a conversation present their concept maps of a topic, exchange them by literally stepping into the shoes of the other and following their reasoning, and eventually coming to an agreement (or at least an agreement to disagree) which forms

the basis for an understanding of the topic addressed. Some kind of knowledge representation then is required as a tool for communication and problem solving. Again, once an expert system has been developed, this knowledge has already been rendered explicit, formal and easily accessible by users.

Finally, in order to evaluate the effectiveness of the embedded helper, on must test the acceptability, effectiveness and usefulness with users. This entails analyzing user achievement and user attitudes toward the expert system. This can then be compared to existing methods of training (seminars, courses, etc.) and of providing assistance (reference guides, user manuals, etc.). This type of continual formative evaluation serves to improve the performance of the expert system as well as its usefulness as a job aid[11].

THE QUARTZ EXPERT SYSTEM EXPERIENCE

QUARTZ[12] is an expert system for the analysis and evaluation of project proposals submitted to a government program for funding. The knowledge and know how of a dozen experts were embodied in this system which not only helps individual analysts evaluate a proposal, but also helps groups of analysts sort out their differences regarding the same project.

A number of unanticipated training needs arose during the development and implementation of this system and these are briefly addressed. There was a unanimous need for additional help in order to use the system in order to better perform their jobs. This comment was as prevalent among the original experts consulted during the development of the expert system, the novice analysts who were just starting to analyze projects, and experts who were not involved in the initial knowledge engineering. The reasons why, however, did show some differences.

The original experts were familiar with the workings of the system but as they did not always have access to a microcomputer, they did not use QUARTZ on a daily basis. Due to this infrequent pattern of usage, there was a need to consult the documentation and ask around before being able to use the system. The users felt that some type of built-in instructions and help facility would be useful. They especially wanted to have comments on the structuring of the knowledge base, examples of analyses (past case studies) and notes on exceptions to the rule, wherever applicable.

Other analysts, who were not part of the original knowledge engineering phase, required more information on the overall nature of an expert system. They needed to know how to begin, how to choose a goal initially (since the system is a backtracking inference mechanism), the implications of possible responses to questions and the general syntax of using the system (for example, how to print out a report). They also needed more help on how to prepare for a session with the expert system. A great deal of the analysts work takes place before they come to consult the expert system, such as meeting the company representatives, obtaining the results of a financial analysis of the company etc.

The novice users, as expected, needed more information on just about everything. Comments would be helpful not only on the structure of the knowledge base but on the actual objects - the terminology used needed clarification. Again, examples of previous analyses would also help serve as a model. While the documentation provided with the system was used, there were a great deal more questions posed on how to get started on the system.

Thus all the users tested on the system agreed that while it was very easy to use, more built in explanations and guidance would be greatly appreciated. In short, while the consultation aspect of the QUARTZ system was fairly transparent to the user, in order for the same system to serve as a job aid, additional features needed to be added. This appears particularly relevant since there have been major changes in the makeup of the team since the beginning of the project, a year and a half ago. This type of high turnover makes it imperative to capture and retain the knowledge and know how of experts, and in turn make this expertise available to newcomers.

The objective of this project was not to develop a teaching or training tool per se. For example, there was no need expressed for a tutorial on decision theory. What was needed was greater guidance on several aspects of the job of analysing project proposals - such as checklists, flowcharts, decision trees etc. The necessary enhancements consist primarily of rendering the reasoning as explicit as possible, explaining each step in detail and allowing a type of advisory "conversation" with the users in order to help ensure they not only know what to do but also why they should do it in the manner prescribed.

CONCLUSIONS

Training has been a problem for business since business began and it will continue to be so. Among today's employees who are still working by the year 2000, fully 75% will require re-training[13]. The development of an embedded training system offer the best long term strategy.

Some of the potential benefits would include:

- to benefit from the experience of others
- to ensure consistent performance
- to decrease the learning time
- to decrease the costs of training
- to increase productivity and job satisfaction
- to increase worker autonomy and self-mastery

The generalization of learned material to the actual job (applications) and the maintenance of skills over a period of time on the job (retention) are also major concerns. The amount of transfer using traditional training is uncomfortably low - only 10%[14]. The transformation of expert systems into job aids will provide employees with on-demand training at their fingertips for the entire time of their employment.

There are many possible uses for an expert system besides the original application it was designed for. It is a relatively simple task to transform an expert system into a job aid by providing adequate instructions and illustrations to show the range of problems the expert system can handle - from the simplest to the most complex. Novice users would have both an expert system and tutor on hand, more experienced users would have an on-line on-call assistant to help with complicated cases, and the expert would have a "master coach" overseeing their work to catch any errors and to suggest improvements.

Our experience with the QUARTZ system suggests that the strategy of conferring some computer-based training characteristics on existing expert systems is a useful short term means of augmenting competence on the job. Expert system technologies are being integrated with computer based learning principles in order to be able to benefit from both technological tools[15]. In this way the use of high technology tools can help teach the skills required of workers in today's increasingly sophisticated and rapidly evolving workplace.

REFERENCES

1. Carr, C. (1988). Making the Human-Computer Marriage Work. Training and Development Journal, May 1988, pp. 65-74.

2. Frenzel, L. (1986). How expert systems can reduce the need for training. In, Proceedings Second Annual Artificial Intelligence and Advanced Computer Technology Conference, Long Beach, CA, 397-402.

3. Hadley, J. (1987). The Expert System Workstation - a Training Alternative. Proceedings, 16th Annual ESD/SMI International Programmable Controllers Conference. Detroit, MI. April 1987.

4. Jones, F. and Lister, R. (1987). The Use of an Expert System Knowledge Base in Training. International Conference on Training for Change - the revolution in commerce and industry. London, U.K. December 1987.

5. Lees, P. (1987). Current Developments in the Application of Advanced Technologies to Training. International Conference on Training for Change - the revolution in commerce and industry. London, U.K., December, 1987.

6. von Foerster, H. (1974). Cybernetics of Cybernetics. Urbana, Illinois: University of Illinois Press.

7. Helfgott, B. (1988). Can training catch up with technology? Personnel Journal, Feb. 1988, pp. 67-72.

8. Kleinschrod, W. (1988). The trend to electronic training. Admin. Management, April 1988, pp. 29-33.

9. Aaronson, A. and Carrol, J. (1987). The Answer is in the Question: a protocol study of intelligent help. Behaviour and Information Technology, 6, 4, 393-402.

10. Pask, G. (1975). Conversation, Cognition and Learning: A Cybernetic Theory and Methodology. Amsterdam: Elsevier.

11. Hofmeister, A. (1986). Formative Evaluation in the Development and Validation of Expert Systems in Education. Computational Intelligence, 2, 65-67.

12. Dalkir, K. and Muzard, J. (1987). QUARTZ: an intelligent assistant for the analysis and evaluation of project proposals. Proceedings, IEEE Montech COMPINT'87 conference. Montreal, Quebec. pp. 208-210.

13. Ross, R. (1988). Technology tackles the training dilemma. High Technology Business, Sept. 1988, pp. 18-23.

14. Baldwin, T. and Ford, J. (1988). Transfer of training: new directions for future research. Personnel Psychology, 41, 63-105.

15. Crews, P. (1987). tbt EXPERT: A Case Study in Integrating Expert System Technology with Computer Assisted Instruction. 3rd International Conference on Data Engineering, Los Angeles, California, February 1987.

LES SYSTÈMES DIAGNOSTIQUES ET L'APPRENTISSAGE DE LA LANGUE

R.H. Brown
R.H. Brown & Associés Inc.
Laval, PQ

Le travail de recherche en intelligence artificielle (IA) appliquée à l'enseignement sans être vraiment récent n'a produit que très peu de produits accessibles en milieu scolaire tel que nous le connaissons aux niveaux élémentaire, secondaire et collégial. Les principaux produits pouvant être répertoriés sont avant tout des outils ponctuels qui aident l'élève-usager à corriger par soi-même son erreur au lieu de donner tout bêtement la réponse exacte après quelques essais infructueux. Pour d'autres produits la notion d'IA désigne un engin navigateur tel que dans le cas de systèmes d'apprentissage sur vidéodisque de niveau 2, 3 et 4 (les systèmes vidéodisques de niveau 0 et 1 ne sont guère plus que des appareils de présentation linéaire).

Par l'emploi de systèmes diagnostiques dans la conception d'outils d'apprentissage, il devient possible de suivre l'élève-usager dans son processus d'acquisition d'une nouvelle notion, de reconnaître les domaines de connaissances mal acquises et intervenir à la source même de l'erreur. Quelques didacticiels de ce genre sont déjà présents sur le marché scolaire. Il est toutefois dommage de noter que trop souvent ces produits sont d'utilisation ardue et de déroulement complexe. Un examen des produits existants sera effectué et largement évalué. Des cas types d'un tel apprentissage seront présentés.

En conclusion, des domaines d'application seront proposés. Quelle sorte d'influence le vidéotexte, et surtout la télématique peuvent-ils apporter aux systèmes diagnostiques et comment, par le mariage du vidéotexte et de ces systèmes, un renforcissement culturel général serait provoqué.

DIAGNOSTIC SYSTEMS AND LANGUAGE LEARNING

Research into the application of artificial intelligence (AI) to education, although not really very recent, has produced very few products available to the educational environment as we know it at the elementary, secondary and college levels. The main products we can list are primarily specific tools to help pupils correct their errors themselves rather than simply giving the correct answer after a number of unsuccessful attempts. For other products, the concept of AI denotes a navigational tool, as in learning systems on videodisk at levels 2, 3 and 4 (level 0 and 1 videodisk systems are hardly more than linear presentation instruments).

By using diagnostic systems in the design of learning tools, we can follow pupils through the process of acquiring new ideas, recognize those knowledge areas that are improperly acquired and take steps at the actual source of errors. A few courseware products of this kind are already on the school market. However, it is unfortunate that these products are all too often difficult to use and complex to follow. Current products will be reviewed and evaluated in broad terms. Typical cases of this type of learning will be presented.

In conclusion, areas of application will be proposed. What kind of influence can videotex, and telematics in particular, have on diagnostic systems, and how can general cultural reinforcement be brought about through the combination of videotex and these systems?

LES SYSTEMES DIAGNOSTIQUES ET L'APPRENTISSAGE DE LA LANGUE

Richard H. Brown

R.H. Brown & Associés Inc.
Laval, Qc.

Le travail de recherche en intelligence artificielle (IA), sans être vraiment récent, n'a commencé à rendre disponibles des applications destinées à l'enseignement qu'au début des années 1980. Alors, ces applications étaient surtout destinées à l'enseignement supérieur et fonctionnait sur des ordinateurs de grande puissance. Seulement quelques années plus tard, on vit apparaître des produits utilisant des notions heuristiques et qui se destineront à l'environnement de la micro-informatique.

Deux types de produits seront alors mis au point. D'une part, des engins navigateurs qui ont pour objet d'interpréter, au moyen de règles, les interventions de l'usager au cours d'une simulation et, d'autre part, des systèmes experts ayant pour but de valider l'acquis de l'usager et en diagnostiquer ses failles le cas échéant. L'apprentissage de la langue se verra particulièrement bien servi par ce dernier, soit les systèmes experts de diagnostique. Au cours du développement d'un tel logiciel appliqué à l'apprentissage du français à l'élémentaire, des observations furent faites quant aux moyens jugés adéquats pour l'intégration du système diagnostique, des paramètres ayant étés utilisés et des restrictions rencontrées.

L'application de l'intelligence artificielle en éducation.

Ainsi que mentionné plus haut, on rencontre deux types d'application de l'IA en formation / éducation: les engins navigateurs (EN) et les systèmes diagnostiques (SD). Les premiers sont utilisés comme support de simulation; en permettant une exploitation interactive de la base de connaissances, les choix, décisions et interventions de l'usager modifient le déroulement de ladite simulation. Bien que l'on retrouve des exemples de produits présentant une interface vidéotex, la plupart son utilisés conjointement avec le vidéodisque. Les EN sont rencontrés en vidéo interactive à partir des niveaux 3 et 4[1] et sont le plus souvent réservés à la formation professionnelle tant en milieu scolaire[2] qu'en milieu de travail[3].

Les SD, si l'on se rapporte à la classification proposée par le Centre APO Québec[4], sont considérés comme étant des systèmes tuteurs. Aussi, bien que partiel, le Centre APO Québec a émis en 1987 un bilan qui indique que la recherche appliquée dans le domaine des logiciels éducatifs de troisième génération[5] s'oriente vers ces systèmes tuteurs[6]. Il est juste de croire que c'est une orientation qui s'est confirmée depuis.

Processus de conception d'un système diagnostique comme outil d'apprentissage destiné au milieu scolaire.

De ce processus de conception, on reconnaît deux domaines distincts. D'une part, les paramètres à prévilégier dans la mise au point d'un SD et, d'autre part, les restrictions externes imposées par le milieu scolaire (le matériel en place et l'environnement de l'usager).

Dans la détermination des paramètres à privilégier on peut se baser sur les critères d'évaluation des logiciels éducatifs tels que proposés par le CMEC[7], c'est à-dire: le contenu pédagogique, le type d'intervention pédagogique, les caractéristiques techniques et le support à l'implantation. Le contenu pédagogique traite d'aspects tels que l'exactitude, l'impartialité et la lisibilité de ce contenu, qui sont mieux adressés de façon pertinente au logiciel mis au point. D'autres aspects tels que l'étendue, la structure et la profondeur de ce contenu doivent être pris en considération dès les premières étapes du design du logiciel. L'application des SD à l'apprentissage de la langue impose le respect de ces trois derniers critères; par la logique sous-jacente aux SD demandant un contenu bien défini présentant d'abord des niveaux généraux, puis, progressant à des niveaux de plus en plus spécifiques. L'ordre de modules notions - exercices ainsi créé permet à l'usager d'acquérir un ensemble de connaissances de base, puis de l'approfondissement de celles-ci grâce à une progression du niveau de complexité des modules exécutés.

Par type d'intervention pédagogique, il est entendu l'interaction avec l'usager, la technique de questionnement, le feed-back, l'évaluation, les méthodes de branchement et les méthodes de contrôle. Ces critères ont pour but de vérifier si les types d'intervention présentés sont appropriés pour l'usager et, deuxièmement, si l'utilisation de l'ordinateur est efficace. Les SD répondent aussi particulièrement bien à ceux-ci. Par sa nature, le SD nécessite une interaction active de la part de l'usager, le contenu peut être présenté seulement en fonction des interventions de l'usager. La technique de questionnement est propre à chaque logiciel développé; toutefois, pour qu'un SD soit fonctionnel, les questions posées doivent être reliées de très près aux notions présentées au préalable. La mise en mémoire des exercises faits permet de s'assurer d'un déroulement non-répétitif des questions posées lorsque le module doit être repris. Une méthode de branchement adapté et correct est de rigueur pour un SD. Suite à une intervention de l'usager, le branchement à être effectué doit d'abord tenir compte de l'intervention proprement dite, mais aussi du contexte dans lequel elle fut faite et de celles qui l'on précédée pour déterminer, s'il y a lieu, toute tendance d'erreur. Bien qu'il y ait des paramètres établis, le feed-back que le SD doit apporter, l'évaluation du rendement en classe du logiciel de même que les méthodes de contrôle du déroulement du logiciel sont des critères discrets - et indépendants du SD lui-même - que le développeur peut déterminer à sa guise. Les deux autres familles de critères à privilégier lors du développement de logiciels destinés au milieu scolaire, soit les caractéristiques techniques et le support à l'implantation, ne sont pas traités ici, n'étant pas du domaine de la présente communication.

Il existe des facteurs externes qui, sans être déterminants du SD lui-même, l'influencent de quelques façons. La puissance des ordinateurs retrouvés dans les écoles de niveau primaire et secondaire, au Québec du moins, sont d'ordinaire des familles 8088 et 68000. Cela a pour effet de limiter la taille de la base de connais-

sances utilisée par le SD si l'on veut que le logiciel reste assez attrayant et dynamique pour soutenir l'attention d'un usager qui réussit à se concentrer relativement peu longtemps. L'utilisation d'une base de connaissances où une synthèse des notions aurait été faite, de même que l'imposition de certaines restrictions quant à l'étendue des interventions de l'usager peut contribuer à éviter une taille trop importante de la base de connaissances et du catalogue d'erreurs, cela sans pour autant porter atteinte à la qualité du SD et du logiciel en général. Le SD devra aussi être adapté à l'environnement dans lequel l'usager doit travailler. Souvent, le nombre d'ordinateurs disponibles est inférieur au nombre d'élèves qui les utilisent et/ou ces derniers sont limités à de courtes périodes de temps d'utilisation. A ce titre, les modules sur lesquels s'articuleront le SD devront être concis et l'usager devra avoir la possibilité d'interrompre sa session et la reprendre plus tard là où il l'avait laissée et ce sans influencer le processus de diagnostique.

Spécificité d'un système diagnostique appliqué à l'apprentissage de la langue.

Lors de la conception d'un SD, il faut tout d'abord déterminer le domaine de la base de connaissances et, à partir de celui-ci, créer un catalogue d'erreurs qui pourra être utilisé pour poser des hypothèses quant aux notions qui auront été mal acquises par l'usager. L'interaction entre cet usager et le SD sera faite à l'aide d'un système-tuteur qui, d'une part, présentera à l'usager de nouvelles informations et lui proposera de nouveaux exercices et, d'autre part, l'informera de la diagnostique des problèmes rencontrés. Lorsqu'un SD est appliqué à l'apprentissage de la langue, sa base de connaissances est composée en premier lieu des règles de grammaire et de syntaxe (il est entendu que l'usager a déjà une bonne connaissance du vocabulaire et qu'il sera utilisé correctement). Le catalogue d'erreurs est, lui, un répertoire des erreurs-type et, respectivement, des facteurs qui peuvent causer ces erreurs-type. Le système-tuteur qui accompagne ce SD interprète, par la suite, l'intervention de l'usager pour en faire ressortir le champ de l'exercice effectué et les caractéristiques de la réponse. Ceci fait, le système-tuteur peut, à l'aide du catalogue d'erreurs, tenter d'identifier l'erreur commise et d'avancer un diagnostique adéquat. Les notions comprises dans la base de connaissances sont présentées d'abord lors de leur apprentissage, puis lorsqu'il y a erreur, les éléments de la notion mal acquise sont utilisés par le système-tuteur pour guider intuitivement l'usager vers une meilleure compréhension de ladite notion.

Influence de la télématique dans l'application des systèmes diagnostiques.

Aidé d'une interface vidéotex adéquate le SD appliqué à l'enseignement, tant en général que pour la langue en particulier, des produits logiciels peuvent être conçus en vue d'une diffusion télématique. Cette diffusion télématique peut avoir lieu à l'intérieur d'un même campus ou communauté grâce à un WAN (Wide Aera Network), par un service modem à modem, ou encore par un service de type ALEX / Minitel ainsi qu'il nous sont proposés respectivement par Bell Canada et CETI.

Ces trois types de diffusion sont comparablement efficaces et s'adressent à des auditoires bien particuliers. Les WAN se retrouveront d'ordinaire dans un milieu défini et l'usager aura à sa disposition une personne-ressource pour combler les vides que le SD n'aura su remplir. Un bon exemple est un campus scolaire ou une

grande institution de ce genre. Un service point à point, ou de modem à modem, peut être retrouvé dans des services d'enseignement ponctuel adressés généralement par des groupes d'intérêt où des communautées définies. Un exemple qui illustrerait ceci serait un système-tuteur greffé à un babillard électronique. Dans les années à venir, ce qui risque de modifier le plus l'enseignement à distance est l'avènement des services télématiques tels que ALEX ou Minitel. Par un terminal vidéotex privé, ou au moyen de son ordinateur personnel, l'usager pourra avoir accès à une vaste base de SD. Cela aura pour effet de rendre encore plus disponible une éducation continue chez des clientèles ayant des horaires restreints. Un domaine d'application de ceci est, par exemple, l'apprentissage et/ou le perfectionnement d'une langue seconde. Déjà, ALEX offre des services d'apprentissage tels que FRSEDUCATE[8].

Conclusion

Les systèmes experts liés à l'enseignement, et plus spécifiquement les SD, deviendront de plus en plus répandus au sein du milieu scolaire. La vitesse à laquelle cela se réalisera dépend de deux facteurs. Premièrement, que les producteurs de logiciel éducatif soient au fait des SD et des systèmes experts en général et, deuxièment, qu'il y ait des efforts constants pour concevoir des logiciels experts qui, en plus de répondre aux critères de qualité d'organismes comme le CMEC, sont ouverts à l'enseignant. On doit espérer aussi pour une rapide popularisation de la télématique de façon à rejoindre le plus grand auditoire possible tant pour diffuser des notions particulières que pour enrichir un fait culturel. Cela s'applique d'autant plus dans des régions dont l'épanouissement culturel est compromis.

Notes

1. L'interactivité de la vidéodisque est divisée en cinq niveaux distincts. Seuls les niveaux 3 (assisté du micro-ordinateur, l'usager peut intervenir dans le déroulement du vidéodisque) et 4 (de plus, l'usager peut utiliser des instruments de simulation) peuvent s'appliquer aux SD. Le vidéodisque au Québec - Etat de la situation et identification des ressources, Centre APO Québec, Montréal (Qc), Octobre 1987. p. 8-9.
2. Projet de vidéodisque de formation en chiurgie dentaire. Mis au point par l'Université de Montréal, cet aide à l'enseignement vise à présenter les techniques et les protocoles utilisés. Le vidéodisque au Québec..., idem, p.38.
3. Vidéodisque de formation "Centrex III". Mis au point par ON-Q Corporation, ce vidéodisque avait pour but d'enseigner au personnel de l'université Concordia l'utilisation du système téléphonique Centrex III de Bell Canada. Ce vidéodisque de niveau 4 utilise le terminal téléphonique comme terminal de contrôle. Le vidéodisque au Québec..., idem, p.42.
4. Chouinard, J., Lafeuille,J., Landreville,G.; Bilan de la recherche et des activités subventionnées dans le domaine des applications pédagogiques de l'ordinateur au Québec de 1984 à 1986. Centre APO Québec, Montréal (Qc). Septembre 1987. p.21.
5. Bilan de la recherche et des activités., idem, p.23.
6. Bilan de la recherche et des activités., idem, p.34.
7. Conseil des Ministres de l'Education (Canada), Evaluation des logiciels - Critères d'évaluation des logiciels éducatifs, Conseil des Ministres de l'Education du Canada, Août 1985. 28p.
8. Bell Canada - Service ALEX, Services offerts sur ALEX. Bell Canada, Montréal (Qc), Janvier 1989. 11p.

CONCEPTION D'UN MODULE D'INTERVENTION D'UN SYSTÈME CONSEILLER

M.-M. Boulet, L. Barbeau, S. Slobodrian
Dép. d'informatique
Université Laval
Québec, PQ

Notre article adresse la problématique de développement d'un module d'intervention prenant place à l'intérieur d'un système conseiller. Dans un premier temps nous décrirons l'univers des usagers d'un logiciel d'application. Nous discuterons ensuite des connaissances requises par le module d'intervention du conseiller et présenterons les principes de psychologie de l'apprentissage à partir desquels nous élaborerons la stratégie d'intervention de celui-ci. Ceci étant précisé, nous présenterons une méthode d'acquisition des connaissances pour le module d'intervention respectant les choix théoriques effectués. Nous illustrerons à l'aide d'exemples le fonctionnement de ce module et nous serons en mesure d'effectuer une démonstration du déroulement d'une séance de travail.
Cette recherche a été rendue possible par l'obtention d'une subvention du Centre Canadien de Recherches sur l'Informatisation des Organisations (CCRIT) et d'un prêt d'appareillage de IBM-Canada. Elle est une partie d'un projet plus global dont les autres partenaires sont: le Conseil national de recherches Canada, l'Université de Leeds et Price Waterhouse.

CONCEPTION OF AN INTERVENTION MODULE OF AN ADVISOR SYSTEM

In this paper, we address the problem of development of an intervention module as part of an advisor system. First, we describe the universe of users of an application software. After, we discuss the knowledge required in the intervention module and present principles of psychology of learning from which the strategy of advice is conceived. Then, we present a knowledge acquisition method for the intervention module, which respects theoretical choices made. We further give examples how the module operates, through a demonstration. This research has received financial support from the Canadian Workplace Automation Research Center (CWARC) and a technical support from IBM-Canada. It is part of a more global project with the National Research Council Canada, the University of Leeds, and Price Waterhouse.

CONCEPTION D'UN MODULE D'INTERVENTION D'UN SYSTEME CONSEILLER

Marie-Michèle Boulet, Ph.D.
Luc Barbeau, Sergio Slobodrian, étudiants gradués

Dépt. d'informatique, Faculté des Sciences et de Génie
Université Laval, Québec, QC G1K 7P4

Au nombre des fonctions prévues à l'intérieur des logiciels d'application, nous avons examiné celle d'aide. Dans la plupart des cas, l'activation de cette fonction déclenche une brève explication sur la syntaxe des commandes. Pour avoir accès à une explication précise, il faut connaître la commande à utiliser pour effectuer une tâche désirée sinon on doit examiner la liste des commandes et, en procédant par essais et erreurs, trouver celle qui convient. Nielsen et al. (1983) et Rosson (1983) suite à une investigation des habiletés d'utilisation par les usagers des logiciels et, en dépit de la disponibilité d'une fonction d'aide, les ont qualifiées de médiocre.

Une solution à ces difficultés serait de pouvoir en tout temps accéder à la connaissance du domaine dans lequel l'usager oeuvre. En fait, la fonction d'aide serait un système à base de connaissances qualifié de conseiller intelligent. Son rôle serait analogue à celui d'un conseiller humain, par exemple d'un professeur qui reçoit un étudiant pour un dépannage. Ainsi, le rôle du système conseiller consisterait à assister une personne sous forme de ressource à la fois passive, pédagogique et active. Sous forme de ressource passive, il répondrait aux interrogations de la personne de façon à lui permettre de compléter sa compréhension d'un phénomène. Sous forme de ressource pédagogique, il enseignerait les concepts ou relations permettant de situer une notion difficile dans le cadre de connaissances préalables. Sous forme de ressource active, le conseiller observerait l'évolution de la tâche et interromprait l'usager, le cas échéant, afin de lui suggérer une action améliorée et peut-être inconnue de l'usager. Le but de notre recherche est de compléter la fonction traditionnelle d'aide d'un logiciel d'application graphique par l'ajout d'un conseiller intelligent en modélisation conceptuelle des données. Cet article adresse l'acquisition et la représentation des connaissances utilisées pour développer le module d'intervention du conseiller auprès des usagers.

Dans un premier temps, nous décrivons l'univers des usagers d'un logiciel d'application. Par la suite, nous présentons le champ d'application du conseiller. Nous expliquons ensuite brièvement le principe de l'outil ayant inspiré notre modèle de représentation des connaissances pour finalement présenter notre méthode d'acquisition des connaissances.

Univers des usagers d'un logiciel d'application.

L'univers des usagers d'un logiciel d'application est constitué par l'ensemble non fini des tâches qui sont réalisables (Figure 1). Cet univers peut ensuite être découpé en sous-univers, un pour chaque groupe d'usagers, ces sous-univers étant identifiés sur la figure 1 par des lettres. On peut considérer par exemple le groupe B comme étant des ingénieurs physiques et le groupe E comme des conseillers en investissement immobilier. Nous remarquons que leurs sous-univers se chevauchent car les tâches effectuées par des deux groupes d'usagers ne sont pas nécessairement disjointes.

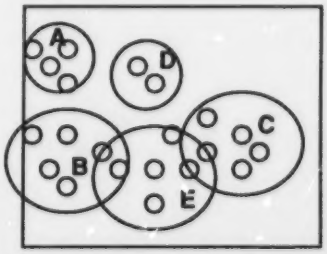


Figure 1. Univers des usagers d'un logiciel d'application.

L'examen de la figure 1 permet également de constater que certains sous-univers ne sont pas entièrement contenus dans l'univers de l'application; c'est l'ensemble des tâches que les usagers croient faisables mais qui ne le sont pas en réalité.

A l'intérieur de chaque sous-univers, on a de petits ensembles de tâches qui sont effectuées par des usagers individuels. Chaque cercle représente l'ensemble des actions qu'un usager doit effectuer pour atteindre son but. C'est là que doit intervenir le conseiller.

Nous définissons le concept de conseiller intelligent comme étant un système informatique, accessible par menu à l'intérieur d'un logiciel d'application et servant à répondre aux questions de l'usager, à lui poser des questions ou à lui enseigner les concepts du domaine dans lequel il oeuvre et à l'intérieur duquel les connaissances sont représentées en termes de structures de préalables.

Champ d'application du conseiller.

Pour être en mesure de conseiller une personne sur la démarche à suivre dans un domaine spécifique, on doit connaître le domaine, pouvoir déterminer le niveau de connaissances de l'usager et posséder des connaissances du processus d'apprentissage (Carroll et al., 1987). Notre conseiller doit intervenir dans le cheminement d'un usager effectuant une tâche complexe, celle de concevoir un modèle conceptuel de données à l'aide d'un logiciel d'application graphique. L'analyse de cette tâche peut être effectuée en adoptant une taxonimie élaborée par un chercheur en psychologie de l'apprentissage. Nous avons retenu celle de Gagné parce qu'il a insisté sur l'analyse des habiletés intellectuelles (Gagné 1968, 1977).

Ainsi, en adoptant la classification des capacités proposée par Gagné (1972) et ses principes de design pédagogique (Gagné et al., 1974), nous associons la tâche de concevoir un modèle conceptuel de données au but d'un cours et considérons celle-ci comme étant une habileté intellectuelle. Un élément important pour Gagné dans l'apprentissage de n'importe quelle habileté intellectuelle est la présence d'habiletés préalables pertinentes (Gagné, 1968). L'importance de son hypothèse de l'apprentissage cumulatif invite à construire des programmes et des plans d'enseignement sous forme de hiérarchie d'apprentissage. Cela consiste à analyser une tâche d'apprentissage pour déterminer les préalables en partant des éléments les plus élevés et en identifiant à chaque niveau les tâches préalables (Gagné, 1977; Gagné et al., 1978).

L'implication du choix de la théorie de l'apprentissage de Gagné pour l'acquisition et la représentation des connaissances de notre conseiller peut à ce point être dégagée. Au lieu de modéliser les connaissances de façon globale en cherchant à établir des liens entre divers concepts, nous les organiserons en termes de structures arborescentes reliées les unes aux autres. Nous acceptons donc que la condition la plus importante pour avoir l'assurance qu'un apprentissage a été fait est l'apprentissage des capacités préalables (Gagné 1968). Nous considérons que chaque personne approche une nouvelle tâche avec une collection différente d'apprentissage d'habiletés préalables et que pour être efficace, un programme d'apprentissage doit prendre en compte ce que la personne sait faire et ce qu'elle ne sait pas faire. Ceci s'évalue par le biais de tests diagnostiques (évaluation formative). Nous voulons donc que notre conseiller puisse positionner une question posée par ou à l'usager ou une erreur de ce même usager à l'intérieur d'une structure de préalables. C'est ici que nous nous inspirons d'un des outils de l'évaluation formative, le test à facettes multiples (Scallon, 1988) pour construire en mémoire d'ordinateur un portrait du profil de l'usager en ce qui a trait à ses difficultés et ce, afin d'être en mesure de lui prodiguer un enseignement correctif. Pour que le système conseiller possède la connaissance du domaine et qu'il puisse associer une question posée à l'usager ou par l'usager à cette connaissance, il faut que lors de l'acquisition des connaissances en vue de procéder à l'élaboration de la base de connaissances, des liens entre les questions et les connaissances soient établis. C'est suivant cette ligne de pensée que nous utilisons certains principes inhérents à l'évaluation formative et à l'enseignement correctif dans le processus d'acquisition des connaissances du domaine.

Le principe du test à facettes multiples.

Le test à facettes multiples permet d'évaluer de façon précise le degré d'atteinte d'une habileté intellectuelle. Il est conçu en vue de la décision soit de faire progresser un élève vers d'autres objectifs, soit de l'impliquer dans des activités correctives spécifiques aux difficultés identifiées. Ce modèle permet, à l'aide de facettes, de définir un ensemble d'items et d'en arriver à un tableau de spécifications à deux ou plusieurs dimensions. Une facette représente toute variation dans les items servant à évaluer le rendement de l'élève.

Au moment de l'acquisition des connaissances pour notre conseiller, nous élaborerons de tels schèmes pour chaque hiérarchie d'apprentissage. Ayant exposé les principes de psychologie de l'apprentissage que nous adoptons, nous détaillons maintenant comment nous procédons pour acquérir les connaissances du module d'intervention du conseiller.

Méthode d'acquisition des connaissances.

A la première étape, le but du module d'intervention du conseiller est formulé. Il s'énonce comme suit: concevoir un modèle conceptuel de données. On procède ensuite à l'analyse de la capacité selon la taxonimie de Gagné (1972): dans notre cas, il s'agit d'une habileté intellectuelle. La seconde grande étape consiste à procéder au découpage du but lié à l'apprentissage. Cette étape s'effectue en suivant la technique proposée par Gagné (1977). Dans un premier temps, l'on procède à une analyse théorique du contenu. Cette analyse s'effectue en plusieurs étapes soit: A- Consultation de manuels spécialisés; B- analyse de plans de cours (ceux de plusieurs professeurs dispensant une formation dans le domaine et ce, pour diverses clientèles); C- analyse du déroulement de cours (film de formation dispensée en milieu scolaire et professionnel). L'on procède ensuite à une synthèse des diverses analyses en vue de formuler d'abord les objectifs d'unités et ensuite les objectifs terminaux de chaque unité. Il n'est pas nécessaire d'avoir effectué l'analyse de tous les objectifs d'unité avant de poursuivre le travail à la phase suivante, ceci en vue d'accélérer le développement d'un premier prototype nécessaire à la validation de l'analyse théorique. Par la suite, on effectue une synthèse des diverses analyses en vue de formuler les objectifs intermédiaires (structure de préalables). A l'aide de la structure de préalables on élabore les schèmes à facettes multiples. Cela consiste à associer les interrogations formulées par ou aux usagers et les explications associées à la (aux) hiérarchie(s) d'apprentissage adéquate(s).

Nature des interventions.

Pour le contenu des interventions, nous avons considéré les observations de divers psychologues de l'apprentissage identifiant quatre différentes structures d'organisation de la mémoire humaine: les réseaux de propositions, les habiletés intellectuelles, les images et les épisodes.

Les réseaux de propositions servent à représenter en mémoire humaine des mots, phrases ou parties de phrases (Kintsch, 1972). Diverses études (Marks et al., 1964; Downey et al., 1968) démontrent qu'une personne a plus de facilité à retenir des phrases lorsqu'elles sont cadrées à l'intérieur d'un contexte significatif. Lors de la conception des interventions touchant la définition de concepts, nous présenterons des définitions à l'usager à l'intérieur d'un contexte ayant pour lui une certaine signification par l'utilisation d'exemples du concept faisant l'objet de définition provenant du modèle conceptuel sur lequel l'usager travaille.

Par rapport aux habiletés intellectuelles, Gagné (1968, 1977) et Weigand (1969) ont démontré que la rétention d'une habileté nécessaire à l'application de règles implique la rétention des habiletés subordonnées. Nos interventions se feront en fonction des connaissances subordonnées non maîtrisées par l'usager et contiendront des mécanismes visant à les détecter et à présenter des moyens pour les corriger.

Gagné et al. (1978) soulignent que les images visuelles sont les plus utiles et proviennent d'objets ou d'événements qui stimulent les sens; une description verbale d'un objet ou d'un événement. peut aussi les générer. Les images ont un effet positif sur le rappel des informations verbales (Paivio, 1975). Des images d'objets concrets (formulaires) seront inclus dans les interventions du conseiller.

Les épisodes, le quatrième type de structure d'organisation de la mémoire humaine, sont utilisés lors de la mémorisation d'événements possédant un ordre chronologique et de nature autobiographique. Les épisodes consistent à pouvoir énoncer "en premier lieu, j'ai fait ceci, ensuite, j'ai fait cela". Le fait d'inclure des épisodes lors du processus d'apprentissage n'a pas permis d'observer d'améliorations significatives de l'apprentissage et de la rétention dans l'énonciation du savoir (Gagné et al., 1978). Pour les habiletés intellectuelles, un nombre égal d'études ont démontré un apport significatif et un apport non significatif. Les interventions du conseiller ne contiendront pas d'épisodes.

L'étape suivante consiste à construire un prototype issu de l'analyse théorique. On entre à l'intérieur de la coquille de système conseiller chacun des éléments de connaissances.Dans notre cas, le principe de séparation des connaissances et des traitements à appliquer a été utilisé.

Les questions posées par ou à l'usager sont situées par le système conseiller dans la matrice représentant le schème pertinent. Le schème se rattachant à un objectif intermédiaire, la valeur d'une cellule de la matrice représentant cet objectif est également changée pour indiquer que l'usager a eu besoin d'aide. L'objectif intermédiaire se rattachant à un objectif terminal, la valeur d'une cellule de la matrice représentant l'ensemble des objectifs intermédiaires est aussi changée. Même processus pour l'objectif d'unité: la valeur d'une cellule de la matrice représentant l'ensemble des objectifs terminaux est changée. Avant de répondre à la question posée par l'usager, le conseiller examine les valeurs de toutes les matrices et selon le nombre de fois auquel l'usager a préalablement référé au même contenu, décide de répondre à la question ou de lui suggérer de revoir certains concepts.

Le prototype sert à valider l'analyse théorique. On filme des usagers élaborant un modèle conceptuel de données à l'aide d'un logiciel d'application auquel la partie du système conseiller à valider est reliée et leur demande de livrer à haute voix leurs réactions. On recueille des questions applicables à la partie de matière que l'analyse théorique a pu ignorer. Des ajustements sont apportés au contenu si nécessaires.

Conclusion.

Nous avons exposé les principes de conception de la coquille du module d'intervention d'un système conseiller; notons qu'un prototype utilisant le schème à facettes multiples comme modèle de représentation des connaissances et de l'usager a été développé dans le cadre d'un autre projet de recherches (Boulet, 1987; Boulet et al., 1987). Dans le futur, nous automatiserons l'acquisition des connaissances en permettant à un expert d'entrer directement les connaissances associées à chacune des facettes. Des cas réels seront donc utilisés. Nous ajouterons un autre module permettant au système de justifier ses interventions.

Remerciements.

Ce projet est subventionné par le CCRIT (Centre Canadien de Recherches sur l'Informatisation du Travail), le Ministère des Affaires Extérieures et IBM-Canada (prêt d'appareillage). Il est une partie d'un projet global impliquant: Le Conseil National de Recherches du Canada, l'Université de Leeds, Price Waterhouse.

Références

Boulet, M.M. "Educational Software Design Using a Diagnostic Approach". Computers and Education. Vol. II , No 3 , Pergamon Journals Ltd., 1987 , pp. 219-227.

Boulet, M.M., Labbé, P., Laprise, Y.,Lemay, F., Lavoie, L. "Conception de systèmes de formation informatisés à base de connaissances selon une approche diagnostique". IEEE: Compint '87. Novembre 1987 , pp. 70-75.

Carroll, J.M., McKendree, J. "Interface Design Issues for Advice-Giving Expert Systems". Communications of the ACM. Vol. 30 , No 1 , January 1987 , pp. 14-31.

Downey, R.G., Hakes, D.T. "Some Psychological Effects of Violating Linguistics Rules". Journal of Verbal Learning and Verbal Behavior. 7, 1968, pp. 158-161.

Gagné, R.M. "Domains of Learning". Interchange. Vol. 3, no. 1, 1972, pp. 1-8.

Gagné, R.M. "Learning Hierarchies". Educational Psychologist. Nov. 1968.

Gagné, R.M. The Conditions of Learning. Third Edition. Holt, Rinehart and Winston, New York, 1977.

Gagné, R.M., Briggs, L.J. Principles of Instructional Design. Holt, Rinehart and Winston, New York, 1974.

Gagné, R.M., White, R.T. "Memory Structures and Learning Outcomes". Review of Educational Research. Vol. 48, no. 2, Spring 1978, pp. 187-222.

Kintsch, W. "Notes on the Structure of Semantic Memory". In E. Tulving et W. Donaldson (Eds.), Organization of Memory. Academic Press, New York, 1972.

Marks, L.E., Miller, G.A. "The Role of Semantic and Syntactic Constraints on the Memorization of English Sentences". Journal of Verbal Learning and Verbal Behavior. 3, 1964, pp. 1-5.

Nielsen, J., Mack, R.L., Bergendorff, K., Grischkowsky, N.L. "Integrated Software Usage in the Professional Work Environment: Evidence from Questionaires and Interviews". Proceedings of ACM CHI 86. April 1986 , pp. 162-167.

Paivio, A. "Coding Distinctions and Repetition Effects in Memory". In G.H. Bower (Ed.), The Psychology of Learning and Motivation. Academic Press, New York, 1975.

Rosson, M.B. "Patterns of Experience in Text Editing". Proceedings of ACM CHI 83. Dec. 1983 , pp. 171-175.

Scallon, G. L'évaluation formative des apprentissages: Tome 1: la réflexion, Tome 2: l'instrumentation. Presses de l'Université Laval, 1988.

Weigand, V.K. A Study of Subordinate Skills in Science Problem Solving. Ph.D. Diss., Univ. of Calif., 1969.

THE EDUCATIONAL POTENTIAL OF RULE-BASED EXPERT SYSTEMS IN THE MICRO ENVIRONMENT

J. Heath, F. Elie, M. Habib
Metavision, Inc.
Montreal, PQ

Three years ago, Metavision began examining the feasibility of developing a knowledge based tutoring shell for use in the micro environment. The final result is the TRILOGY system, a combined expert system shell, intelligent tutor and simple authoring system, written in ProLog and running on a standard IBM-PC.

The expert system is designed to handle small knowledge bases (200-500 rules). Consultations with the expert are saved to produce case studies for use by the tutor. Using this information, the tutor builds mixed-initiative dialogue with a student, using modelling techniques to produce an optimal level of discussion. To overcome difficulties produced by any discrepancy between the problem-solving methods used by the computer and a human expert, a teacher's toolkit allows the instructor to edit the solution tree of a case scenario, and integrate text into the tutorial dialogue.

Several features of advanced artificial intelligence systems have been successfully implemented, including a sophisticated expert reasoning explanation module, student initiative in tutorial dialogue, computer-generated student evaluation methods, accessible menu-based interfacing for all operations, natural-language output and consistency-checking for rule-base development.

POTENTIEL PÉDAGOGIQUE DE SYSTÈMES EXPERTS À BASE DE RÈGLES POUR MICRO-ORDINATEUR

Il y a trois ans, Metavision a commencé l'étude de la faisabilité d'un intégrateur tuteur cognitif qui tournerait sur un micro-ordinateur. Le résultat de ces efforts est le système TRILOGY rédigé en ProLog et exécutable par un IBM-PC, soit un intégrateur comprenant un système expert, un tuteur intelligent et un système auteur rudimentaire.

Le système expert est en mesure de gérer de petites bases de connaissances (200 à 500 règles). Les consultations avec l'expert sont sauvegardées en vue de constituer des études de cas pour le tuteur. À partir de ces informations, le tuteur amorce un dialogue à initiatives mixes avec un étudiant en recourant à la modélisation pour obtenir un niveau d'échange optimal. Pour surmonter toutes les difficultés inhérentes à la différence entre la démarche de résolution des problèmes d'un ordinateur et celle d'un humain, l'instructeur dispose d'une boîte à outils qui lui permet d'éditer l'arbre de solution d'un scénario type et d'intégrer du texte dans le dialogue tutoriel.

Plusieurs fonctions des systèmes d'intelligence artificiel avancés ont été implantées avec succès, notamment un module expert sophistiqué d'explication des raisonnements, la gestion des initiatives de l'étudiant dans le dialogue tutoriel, des méthodes d'évaluation de l'étudiant générées par la machine, un interfaçage de toutes les opérations facilement accessibles avec un menu, l'expression des résultats en langage naturel et le développement à partir de règles de la vérification de la cohérence.

THE EDUCATIONAL POTENTIAL OF RULE-BASED EXPERT SYSTEMS IN THE MICRO ENVIRONMENT

Joe Heath
Frederic Elie, Michel Habib

Metavision, Inc.
307 Ste-Catherine O, Suite 720, Montreal PQ, H2X 2A3

For the past three years, Metavision has been developing a combined expert-tutoring system called TRILOGY. Overall, the product can be described as a structured educational environment designed to exploit the pedagogical potential immanent in expert system knowledge representation. In the final implementation, TRILOGY has three components: an expert system shell which provides fully descriptive and coherent explanations of expert behaviour, an intelligent tutoring system which allows the student to try solving sample problems, and a limited authoring system which the instructor can use to integrate additional text into the tutorial dialogue.

The TRILOGY system falls into the category of a small knowledge system shell, designed to accept 200-500 rules, and running on a standard IBM-PC. Overall, the project was aimed at bringing some of the basic components of intelligent tutoring — student modelling, automatic discourse generation and interactive mixed-initiative dialogue — to the micro expert system user. In the process, some of the central aporia of shell tutoring were confronted from a highly pragmatic perspective: the tentative solutions implemented are presented herein.

When expert systems first became a popular software item, they were primarily integrated as problem-solving tools in the immediately productive work environment. Above all, expert systems were built and used to produce the right answer. But it soon became evident that the structure of knowledge representation within these systems lent itself to other applications.

In an expert system shell, the knowledge of the actual domain is entered by the user, hence is completely seperate from the inferencing system. In other words, with respect to a particular problem to be solved, the 'how' is seperate from the 'what'. In most small expert systems, this 'what' is a series of concepts, procedures, rules-of-thumb, etc., coded into a standard format of knowledge representation — if-then rules.

Once a generic representation of knowledge is developed, it becomes possible to use this knowledge for more than simple problem-solving. A completed solution tree can be used as the basis of discussion with a student independent of the expert system — as a 'scenario' for tutorial dialogue.

Much of this is rehash, since research work on intelligent tutoring companions to expert systems dates back to the late 1970's. However, much of this research became bogged down in largely academic considerations, centered mainly around problems with student modelling and dialogue strategies. Since then, solutions to these problems have generally been acquired by abandoning the notion of a 'shell' and producing dedicated systems. This ignores the fact that while systems like GUIDON (developed to accompany the MYCIN rule base) had their obvious difficulties, in general the advantages far outweighed the drawbacks.

The problems with shell tutors are quite real, but their importance is generally proportionate to the size of the rule base under consideration. Similarly, the margin of acceptable error permitted to a tutor varies inversely with the size of the rule base. Working in the micro environment, it was possible to set aside some of the dicier aspects of tutoring shells and focus on producing a functional implementation.

The TRILOGY system, based loosely on GUIDON, attempts to reproduce the basic strengths of that system, while offering the tools necessary to cope with persistently problematic areas.

The Tutor

Using this tutoring methodology, a lesson is constructed using the expert system. The instructor answers the expert's questions, and then saves the completed solution tree on disk. When a tutoring session begins, the tutor loads in a selected solution tree to be used as a case study problem for the student to solve.

During a consultation with the expert system, the computer knows how to solve a problem, and the user knows all the case information. The computer asks the user for what it needs, then works out a solution. In a tutoring session, the computer has all the case information, and the student must 'play expert', by asking the system for the basic information, then constructing a series of hypotheses leading to the final solution.

When the student asks for the value of a particular object, the tutor will provide it to him/her iff the expert also had to ask for it. If the expert used rules to conclude the value, the object will become a topic of conversation. The student will then have to ask for the information needed to apply the rules. Once this has been gathered, the student will be expected to make an hypothesis about the topic. To evaluate the student's knowledge, the tutor also constructs quizzes using the rule base and solution tree.

For some rule bases, this basic tutoring method works without a hitch. But more often than not, there will be problems. Some of these can be overcome directly, others simply have to be avoided through higher quality knowledge engineering. Some of the smaller problems which arise are dealt with as follows:

- The rule base may contain technical vocabulary and keywords which a student cannot be expected to know: All value keywords, object-attribute titles, and facts can be given complete natural-language translations.

The user interacts with the system through menus constructed with these phrases.

- The tutor can easily show how rules are used, but cannot explain concepts: TRILOGY contains a small authoring system which allows an instructor to 'link' segments of text to any rule, object, or solution tree. This text will then be integrated automatically into the tutorial dialogue. Thus, when the tutor discusses a goal which corresponds to a technical concept the student might be unaware of, it can present a prepared introduction and conclusion. This is all done in a 'non-intelligent' fashion, allowing the instructor complete control over all such presentations.

- Some rules are strictly procedural, they have no conceptual value. But the tutor cannot tell these apart from other rules: The system contains a toolkit which allows the instructor to 'edit' the solution tree in a variety of ways, including cutting out rules not directly implicated in the solution tree. Once a rule is cut, it will remain in the rule base, the student can examine it, but the tutor will not discuss or quiz on it. (This is handy for eliminating unfired 'default' rules) Rules can also be 'turned off' before a session with the expert system begins. If required, explanations can be included local to the edited solution tree, associated with cut rules, explaining to the student what has transpired.

- Some entire subgoals have no conceptual value. Although they are concluded by rules, they are not worth discussing with a student: The instructor can also cut the entire solution tree at a particular branch. In such cases, any subgoals and rules 'below' the cut point are eliminated, and the cut point object is considered to be a piece of case data. This effectively eliminates the object from discussion. Text can be added to the cut point, informing the student of the abbreviation.

Aspects of Knowledge Engineering

There are problems which arise which cannot be dealt with so easily. These are related to either information paucity or overload:
- Solution tree lacks depth: There are not enough subgoals, some concepts are omitted. For instance, a set of rules may conclude a prescription directly from observation, omitting the diagnosis. There is no way the tutor can substitue in this missing information.
- Solution tree lacks breadth at the higher levels: This means that the subgoals closest to an including the main goal are too general, and have a very high rule/object ratio.

This is where the architecture of the rule base becomes important to the coherency of the tutorial dialogue. Neither of the above two problems are fatal, they simply marr the tuor's performance and confuse the student. In the first case, although the student may be able to produce the right answer, like Searle's Chinese room, s/he does not actually 'know' how to solve the problem. In the second case, specific diagnoses which are conceptual extremes will often be discussed within the framework of a single object, generally confusing the student.

Thus, for the TRILOGY tutor to function most effectively, there is a specific style of knowledge engineering which avoids these pitfalls, yet maintains diagnostic integrity and economy. The overall objective is to produce a solution tree with an ideally low number of rules per object. This can be achieved through two basic tactics:
- From the top of the solution tree to the bottom, having the backward-chaining mechanism moving downwards through formally schematized systematic levels of generalization with respect to the main goal. The basic idea is to create several layers of intermediate problems which organize the case information according to a conceptual hierarchy.
- Breaking up large or important objects — instead of having a general object with thirty possible values, have three objects with ten possible values each.

This approach may seem counterintuitive to some knowledge engineers, who, as programmers, tend to place economy over clarity. In this case, they are being asked to make a programme larger, not smaller. If the internal structure of a knowledge base is viewed strictly instrumentally, i.e. as the means to an answer, this makes no sense. But for the tutor, the main goal is not the most important feature of a solution tree -- every goal is important.

A Sample

Simple organizational tactics such as those outlined above can greatly clarify the system knowledge for a student and make the tutor a more effective pedagogical tool. Transcripts of a few sample consultations below should better illustrate the specific impact of these considerations on the quality of tutorial dialogue.

The fragments below are edited segments of a tutorial dialogue based on the analysis of a single patient by two very small (30 rules) medical rule bases. It is important to note that the information, both facts and rules, in both systems is virtually identical. Both consultations are based on the same patient -- i.e. the same symptoms, the same case data. Both rule bases also produced the correct conclusion, within 5 certainty points (out of 100) of each other. The only difference is in how the rules organize information.

Both rule bases perform a pre-clinical diagnosis of stomcah problems. The first is well organized. It contains four goals, and the main diagnositic problem is broken into two goals, as outlined in the subgoal tree sketch at the beginning of the consultation. The second rule base contains all the same information, but has only two goals. The diagnostic goal has been merged into a single category, and an intermediate subgoal has been eliminated. The consultation generated with this rulebase is technically correct, it just fails to present the information as clearly as the first.

Note that the dialogue below is all computer-generated. The solution tree has not been edtied, and no text has been added using the authoring system.

RULEBASE — CLEAN

We are now discussing the main goal of the consultation
In order to determine the type of exam required,
we will have to resolve the following problems:
1. type of exam
    1. physical problem
        1. evidence of an ulcer
    2. problem related to infection

We will discuss all the rules which were used by the

expert to conclude about the current topic, starting with rule R3.ex...

Let's examine the antecedents of this rule.

We still have to determine physical problem
You should now proceed to ask questions which will enable you to make an hypothesis about this goal

Using rule R2.U,
assuming
there is evidence of an ulcer (certainty 50) and
subject is repelled by cigarettes (certainty 60) and
subject finds meat distasteful (certainty 60),
does this tell you that
physical problem is not benign ulcer (certainty 50)
» yes
Correct

Using rule R1b.U,
using,
there is evidence of an ulcer (certainty 50) and
the current season is winter (certainty 100),
one can conclude that:
1. physical problem is hernia (certainty 20)
2. physical problem is not benign ulcer (certainty 25)
3. physical problem is malignant ulcer (certainty 30)
4. physical problem is perforation (certainty 80)
» 3
physical problem is malignant ulcer
Right
You forgot to mention physical problem is not benign ulcer

The topic of our discussion is now physical problem
Here are the factors you should consider:
type of pain
frequency of pain
difficulty breathing
subject repelled by cigarettes
subject repelled by meat
evidence of an ulcer
current season
lack of energy
symptoms of anemia
There are 10 rules used to establish conclusions about this topic. Here are the possible values:
physical problem is hernia
physical problem is benign ulcer
physical problem is perforation
physical problem is malignant ulcer

» case
Please select condition from menu:
» exposure to a contagion suspected
This relates to a previous subgoal: type of exam
exposure to a contagion is no (certainty 100)

» discuss condition
Please select condition from menu:
» evidence of an ulcer
As you like...

Using R3.U,
assuming
subject is suffering from back pains (certainty 60)
does this tell you that
there is evidence of an ulcer (certainty 20)
» no
Wrong, for this rule the fact supports the conclusion
The topic of our discussion is now evidence of an ulcer
» case
Please select a condition from the menu:
» subject is under stress
Yes, the subject suffers from stress (certainty 100)
*** The expert has made a conclusion about evidence of an ulcer
» hypothesis
You should now enter an hypothesis for the value of evidence of an ulcer. Note that there is only one correct answer
» there is evidence of an ulcer
Consider:
presence of back pains is yes (certainty 100)
What does this tell you about evidence of an ulcer
» there is evidence of an ulcer
You're right, and your hypothesis is correct
The exact value is there is evidence of an ulcer (certainty 85). Now back to physical problem...

» tell me
physical problem is malignant ulcer (certainty 95)
Would you like to see how the expert was able to reach this conclusion?
» no
Okay, back to type of exam
We have virtually exhausted our discussion of this goal, would you like to make an hypothesis?
» no
Okay, we still have to discuss the following subgoal:
problem related to infection

(etc. etc.)

RULEBASE -- MUSH

We are now discussing the main goal of the consultation
In order to determine the type of exam required,
we will have to resolve the following problems:
1. type of exam
1. diagnosis of problem

We will discuss all the rules which were used by the expert to conclude about this goal, starting with rule R1.ex...

Let's examine the antecedents of this rule.
We still have to determine diagnosis of problem
You should now proceed to ask questions which will enable you to make an hypothesis about this goal

Using rule R1b.U,
assuming
the subject suffers from stress (certainty 50) and
the type of pain is local (certainty 60) and
the pain increases before meals (certainty 60) and
the frequency of pains is constant (certainty 60) and
presence of back pains is yes (certainty 60)and
subject is repelled by cigarettes (certainty 60) and
subject finds meat distasteful (certainty 60),
does this tell you that
diagnosis of problem is benign ulcer (certainty 50)
» no
Correct

The topic of our discussion is now diagnosis of problem
Here are the factors you should consider:
type of pain

frequency of pain
difficulty breathing
subject repelled by cigarettes
subject repelled by meat
current season
lack of energy
symptoms of anemia
exposure to contagion suspected
subject is under stress
presence of back pains
when the pain increases
subject has a fever

There are 18 rules used to conlude about this topic. Here are the possible values:
diagnosis of problem is hernia
diagnosis of problem is toxic inflammation
diagnosis of problem is malignant ulcer
diagnosis of problem is benign ulcer
diagnosis of problem is general inflammation
diagnosis of problem is perforation
diagnosis of problem is local inflammation
diagnosis of problem is indigestion

» case
Please select condition from menu:
» subject repelled by cigarettes
subject is repelled by cigarettes (certainty 100)

» case
Please select from menu
» exposure to a contagion suspected
exposure to a contagion is no (certainty 100)

» case
Please select a condtion from menu:
» presence of back pains
subject is suffering from back pains (certainty 100)
Using rule R2.U
assuming
subject is typically nervous (certainty 60) and
the type of pain is local (certainty 60) and
the pain increases before meals (certainty 60) and
the freqency of pains is constant (certainty 60) and
presence of back pains is yes (certainty 60) and
the current season is winter (certainty 60),
does this tell you that
the diagnosis of the problem is benign ulcer (certainty 30)
» no
Correct
» tell me
the diagnosis of the problem is malignant ulcer
the diagnosis of the problem is not toxic inflammation
Would you like to see how the expert arrived at this conclusion?
» no
Okay,
***The expert has make a FINAL conclusion about type of exam
You should now enter an hypothesis for the value of type of exam

(etc. etc.)

Obviously, the first rule base generates a much cleaner, more comprehensive dialogue than the second. Unfortunately, most rule bases are written more like the second than the first. But making tutors more intelligent is not necessarily the answer, and the shell concept should not be dismissed offhand. Very simple AI tools can can perform at the level of more advanced systems IF the knowledge provided is sufficiently clear.

Furthermore, it seems probable that as expert systems become more an more common, the requirements of system maintenance and upgrading will call for a higher level of conceptual clarity.

TRILOGY is of course not a research project, but a piece of commercial software. However, it may produce important spin-offs, particularly because it should establish a large body of feedback information from students working in signfficantly different domains, from the classroom to the work place. Hopefully this information will be useful not only for the development of more sophisticated tutoring shells, but also as a set of rough guidelines for improving knowledge representation.

Future Directions

At the comment, plans for improvement on the system extend in two different directions:

- Implementation of a (currently partially developed) natural language parser and language environment editor, allowing the user to dispense with the menu system entirely.

- Creation of an 'ideal type' modelling system for knowledge development consistency checking which could in turn be used to refine tutoring dialogue strategies and modelling techniques. The basic thrust of this would be to produce a 'shadow' quasi-semantic net which the tutor could use to move beyond simply teaching rules and start dealing with object concepts on a meaningful level.

STRUCTURE DES BASES DE CONNAISSANCES D'UN SYSTÈME TUTORIEL INTELLIGENT POUVANT ÊTRE UTILISÉ POUR L'APPRENTISSAGE DE LA MUSIQUE

R. Tremblay
Dép. des sciences économiques et administratives
Université du Québec à Chicoutimi
Chicoutimi, PQ

B.J. Fugère, L. Niem
Royal Military College of Canada
Kingston, ON

Dans cet article, les auteurs essaient de voir si la musique peut être un domaine d'étude valable pour l'élaboration de systèmes tutoriels intelligents. La principale raison de ce choix est que d'une part, la musique est un langage bien défini et que d'autre part, elle constitue l'une des activités créatrices les plus fondamentales.

Pour illustrer ces possibilités, on envisage l'utilisation d'un système tutoriel pour enseigner certains aspects de la musique. Pour être en mesure de développer un système pour différentes applications (éducation de l'oreille, théorie musicale, harmonie, histoire...), il serait grandement utile de créer un modèle qui structurerait l'acquisition et le fondement de la connaissance. Ce modèle est présenté, puis utilisé pour élaborer des programmes dans les domaines de l'éducation de l'oreille et de la théorie musicale. Les problèmes de sa mise en application sont expliqués ainsi que les particularités de son application.

Ce projet fait partie d'un projet de recherche d'envergure, le projet HERON qui a été mis sur pied à l'Université de Montréal (Frasson 1988).

THE STRUCTURE OF THE KNOWLEDGE BASES OF AN INTELLIGENT TUTORING SYSTEM TO BE USED TO LEARN MUSIC

In this paper, the authors discuss the relevance of using music as a domain of study for developing intelligent tutoring systems. The main reason for this choice is the fact that music is a well defined language and, on the other hand, it is also one of the most fundamentally creative activities.

To illustrate these possibilities, a tutoring system to teach some aspects of music is considered. In order to be able to develop such a system for different applications (ear training, music theory, harmony, history ...), a model to structure the knowledge acquisition and the knowledge bases involved is highly desirable. Such a model is introduced and then applied to develop programs to teach ear training and music theory. The problems of implantation are discussed and finally the particularities of this application are mentioned.

This project is part of a larger research project, the HERON Project which has originated at the University of Montréal (Frasson 1988).

STRUCTURE DES BASES DE CONNAISSANCES D'UN SYSTEME TUTORIEL INTELLIGENT POUVANT ETRE UTILISÉ POUR L'APPRENTISSAGE DE LA MUSIQUE

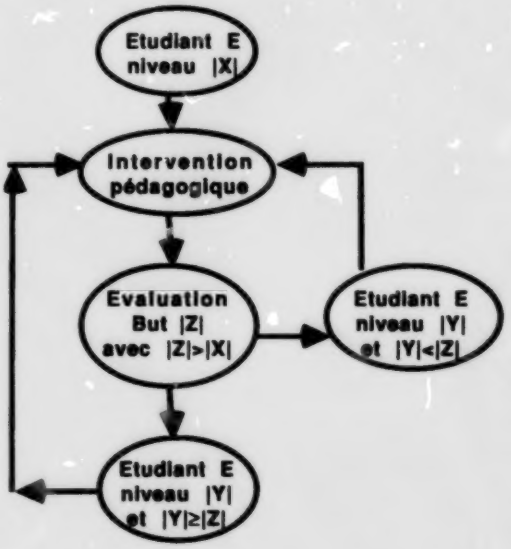
R. Tremblay[1], B.J. Fugère[2] et L. Niem[2]

[1]Université du Québec à Chicoutimi
[2]Royal Military College of Canada

1.0. Introduction: Systèmes tutoriels. Un système aidant une personne dans l'apprentissage d'un sujet donné est appelé un système tutoriel. Si ce système utilise en plus les techniques du domaine de l'intelligence artificielle, on dit qu'il devient un système tutoriel intelligent [2]. Plus encore, si le système peut tenir compte de paramètres mesurant le profil psychologique de l'usager dans le processus d'apprentissage qu'il utilise, nous le qualifions de système tutoriel intelligent et adaptatif *(Adaptative ICAI or Intelligent Computer Aid Instruction)*. Le système considéré dans cette communication fait partie de cette catégorie.

Un système tutoriel intelligent adaptatif tient compte de l'apprenant pour déterminer l'environnement et définir le micro-monde dans lequel il intervient. En situation effective d'apprentissage, il est essentiel d'attacher autant d'importance au type de l'étudiant en processus d'apprentissage qu'aux connaissances de celui-ci du sujet enseigné [2]. Le système tutoriel enrichi doit être alors muni d'un processus d'évaluation beaucoup plus élaboré. Ce dernier, prenant en charge multiples paramètres, mesure dans son ensemble la situation d'apprentissage pour permettre la prise de décision.

Dans le diagramme qui suit, l'intervention pédagogique représente des activités suggérées ou imposées par le système à l'étudiant et l'évaluation se fait selon plusieurs points de vue: usager, objectifs, programme, activités, etc... [7]. Dans ce diagramme, |W| représente la norme de W, W étant la mesure du progrès accompli par l'étudiant ou désiré au moment considéré. La norme est fonction du sujet enseigné. Une façon d'approcher ce problème a été d'utiliser un modèle déjà développé pour décrire et simuler une situation d'apprentissage [7]. Ce modèle est particulièrement intéressant parce qu'il possède les propriétés d'auto-évaluation et de rétro-action, essentielles pour la plupart des applications, surtout celles faisant intervenir le système de valeurs des intervenants: émotion, croyance, jugement, etc... Parmi ces sujets, on trouve sans conteste le domaine de la musique, celui des langues, de la composition, de la peinture, et bien d'autres encore.



3.0. Le modèle. Essentiellement, le modèle fait intervenir quatre différentes bases de connaissances interreliées que l'on nomme programme (P), contexte (C), activité (A) et résultat (R),(Voir le schéma suivant). Les lettres α, β, γ et δ, de même que U, V, X, Y et S, désignent les zones d'interaction entre les bases de connaissances. Les règles contenues dans les zones α, β, γ et δ ont essentiellement la même structure, de même pour U, V, X et

Y. La zone S contient l'ensemble des règles de commucation avec l'extérieur en cas de conflit. Par exemple, la zone α dans un premier temps assure la compatibilité entre les bases Programme et Contexte. Si l'incompatibilité est démontrée, il passe le contrôle à la zone S. Dans le cas contraire, des règles déterminent une activité de la base A. Les résultats escomptés sont déduits par des règles de niveau supérieur contenues dans la zone U. La structure et le rôle des autres intersections sont similaires. Plus précisément, on trouve:

- le **niveau 0** (P, C, A et R) ne contenant que des faits constitués de caractéristiques et de normes,
- le **niveau1** (α, β, γ et δ), formé de règles qui vérifiront la cohérence des faits du niveau 0 relativement aux bases prises deux à deux, ou produiront si possible et si nécessaire d'autres faits au niveau 0 dans les bases concernées ou encore questionneront le niveau 2,
- le **niveau 2**, formé de règles qui utiliseront les faits du niveau 0 pour les modifier si possible et si nécessaire en considérant les bases prises trois à trois ou encore questionneront le niveau 3 et finalement,
- le **niveau3**, intervenant en dernier recours lorsque le système ne peut résoudre le problème décelé. Il identifiera l'expert humain qui doit intervenir, lui présentant un éclairage de la situation, lui proposant des voies de solutions et fera les changements des faits ou règles proposés par l'expert du domaine ou les experts impliqués si le problème touche plusieurs bases.

Un exemple de fonctionnement de ce modèle est illustré par la suite d'actions et de règles suivantes:

Action 1
Identifier l'usager (paramètres d'identification)
Règle 1
Si l'usager est connu (Le contexte global noté C est défini)
Alors Continuer
Sinon Déterminer un contexte global C
Déterminer un programme global noté P (en rapport à un objectif général)
Règle 2
Si P et C sont compatibles
Alors Continuer
Sinon Vérifier C et P
(Le système vérifie si l'usager tel que connu est en mesure d'aborder le programme proposé. S'il décèle une anomalie, il remettra le contexte en question. Si le contexte semble correct après vérification, il remettra le programme en question en l'élargissant si possible. Si cela s'avère impossible, il questionnera le superviseur S)
Action 2
Déterminer p(usager), un programme local à l'usager noté p.
(Il s'agit de positionner l'usager dans l'espace programme ou programme global P (en rapport à un objectif spécifique))
Action 3
Choisir un c(usager), un contexte local à l'usager noté c.
(Le contexte local c est un sous-ensemble du contexte global C. Il pourrait être à la limite précisé par le système afin d'optimiser la situation d'apprentissage à l'usager au moment présent.)
Action 4
Déterminer A(usager), un micro-monde d'activités ou une activité globale notés A .
(Chercher et présenter à l'usager un choix d'activités spécifiques propres au contexte local c et au programme local p de l'usager)
Action 5
Déterminer a(usager), une activité locale à l'usager notée a
Règle 3
Si a(usager) existe
(Si l'usager fait un choix d'une des activités présentées)
Alors Continuer
Sinon Déterminer un contexte global C en consultant le résultat global R précédent et en tenant compte du programme global P en cours.
(Le système changera possiblement les normes à considérer)
Retourner à l'action 2
Action 5
L'usager réalise l'activité choisie.
(Le système perçoit les paramètres actuels)
Action 6
Déterminer R(usager), un résultat global à

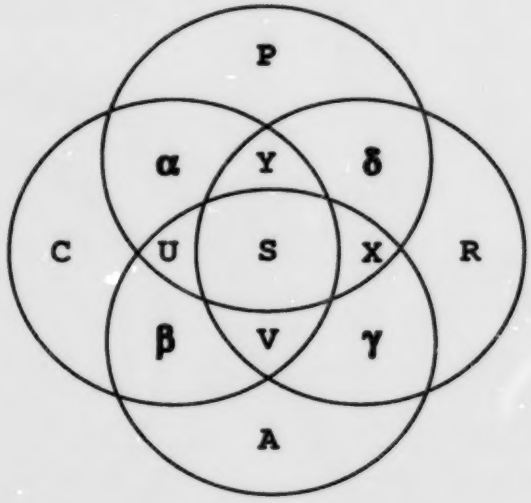
l'usager noté R

(Le système établit un ensemble de normes et de règles d'évaluation cohérentes en regard des valeurs des paramètres perçus lors de l'exécution de l'activité)

etc...

Graphiquement, on peut représenter ces bases de connaissances comme suit:

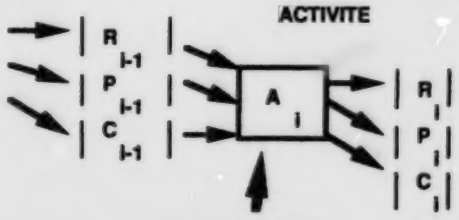
Modèle



3.1. **Programme**. Le programme contient les différents concepts que l'étudiant devra maîtriser. Ce savoir est résumé dans des objets structurés formant les noeuds d'un réseau sémantique. Cette approche permet une représentation de différents niveaux, de prérequis, d'options, etc... . Les objets sont structurés en éléments atomiques. On trouvera une description de la structure de cette base de connaissances dans [6].

3.2. **Contexte**. La base de connaissances identifiée contexte contient toute l'information concernant l'apprenant. Cette information se divise en deux grandes catégories: les connaissances acquises de l'usager relatives à un programme ou sujet donné et le type de l'usager en fonction du temps. La première partie de l'information est relativement facile à acquérir. Fondamentalement, elle se résume à la liste des concepts maîtrisés par l'usager et le cheminement suivi, ces concepts étant ceux décrits dans la base de connaissances programme. Évidemment, il est toujours possible d'attacher des attributs à chacun d'eux permettant de réfléter le degré de maîtrise du concept. Tant qu'au type de l'usager, il est plus difficile à caractériser et à quantifier. Une façon possible de résoudre ce problème est offerte par l'approche de la programmation neurolinguistique [1]. Cette approche fournit les moyens qui permettent d'identifier et de caractériser les différents groupes d'individus conduisant entre autres à la sélection des moyens de communication les plus appropriés.
La connaissance de l'étudiant est représentée en fonction de trois catégories: état, évènement et action. Les états sont décrits en termes de concepts, les actions décrivent la transition d'un état à l'autre et les évènements décrivent les conditions nécessaires pour qu'une transition ait lieu.

3.3. **Activité**. Les diverses activités sont représentées par un n-tuplet et d'un ensemble de règles d'utilisation. Les composantes de vecteur sont des arbres dans lesquels les activités sont les feuilles et les noeuds représentent les paramètres nécessaires à identifier et activer l'activité. Cette structure nous donne la possibilité de choix multiples d'activités pour un contexte donné. On a:



où l'action-usager représente l'interaction entre l'usager et l'activité en cours. Le résultat de cette intervention est un nouvel état de contexte, de programme et de résultat pour l'usager et ces nouvelles valeurs servent à déterminer l'activité qui suivra.

3.4. **Résultat**. Pour un apprenant donné, cette base de connaissances contient son évaluation et celle du système. Encore une fois, l'évaluation est représentée sous forme d'un n-tuplet et d'un ensemble de règles. Ces règles gouvernent le processus d'évaluation.

L'ensemble des paramètres utilisés pour l'évaluation sont:

(usager, type, programme, activité)

où ces paramètres, à l'exception du premier, représentent eux-mêmes le résultat d'une évaluation.

3.4.1. **Règles d'évaluation**. Au niveau 1, une activité est évaluée en examinant un vecteur résultat $\mathfrak{R} = \{ r_1, r_2, ..., r_n \}$. Cela est représenté par la zone $\gamma(A,R)$ dans le modèle. Au même niveau de règles, nous avons également la relation $\delta(R,P)$ tenant compte de l'évaluation du programme par rapport au vecteur résultat $\mathfrak{R}$. Au niveau 2, nous retrouvons des fonctions comme U(A,R,P) laquelle évalue le contexte considéré comme une fonction des activités, des résultats obtenus, du programme suivi, etc... Au troisième niveau, une évaluation globale est faite se servant des évaluations des niveaux précédents. Par exemple, les décisions suivantes sont prises à ce niveau:

"L'apprenant peut-il avancer dans le programme en cours?"

"Le contexte doit-il être modifié?"

"Le programme doit-il être changé?"

etc...

4. **Conclusion**. Plusieurs problèmes restent à définir et à résoudre avant d'avoir un système tutoriel intelligent pour l'enseignement (de la musique par exemple) ayant toutes les caractéristiques désirées. En ce moment, un prototype est en phase de réalisation utilisant le système auteur PC-PILOT, PROLOG et une interface intégrant différents médias (vidéodisque -voix-son-image) laquelle est déjà développée [4], [5].

L'utilisation de l'intelligence artificielle pour l'apprentissage est certainement réalisable dans un proche avenir. Ce sujet fait intervenir différents aspects de la science technologie, de la psychologie, de la pédagogie et constitue un véritable défi. Mais pour espérer un tel succès, les principes de base gouvernant les processus impliqués devront être découverts et mieux compris.

REFERENCES

[1] Bandler, R., Grinder, J., The Structure of Magic, a Book About Language and Therapy, Science and Behaviour, Palo Alto, Vol.1, 1975, Vol.2, 1976.

[2] Clancey, W.J., Knoledge-Based Tutoring: The Guidon Program, MIT Press, 1987.

[3] Fugère, B.J., Tremblay, R. and Geleyn, L. , A Model for Devoloping a Tutoring System in Music, Proceeding of AAAI-88, First Workshop on AI and Music, St.Paul, Minnesota, USA, 1988.

[4] Girard, P., D.A.V.O.S., a Computerized Videodisc Animation with Synthetized Speech, Proceeding IEEE Graphics Interface, Edmonton, Canada,1988.

[5] Girard, P. and Gecsei, J., D.A.V.O.S., a Real-Time Speech, Visual Computer, 1989 (to appear).

[6] Ouellet, A., L'évaluation créative: une approche systématique des valeurs, Presse de l'Université du Québec, Québec, Canada, 1983.

[7] Proceeding IST-88, Montréal, UdeM, June 1988.

[8] Road Curtis, Research in Music and Artificial Intelligence Computer Surveys, Vol.17, No.2, June 1985.

EMMA: AN INTELLIGENT TUTOR

M.T. Quigley
Dept. of Curriculum and Instruction
Memorial University of Newfoundland
St. John's, NF

EMMA is a computer tutor for algebra designed to overcome the three main deficiencies of educational software and to run on microcomputers found in schools. EMMA consists of four linked expert systems. The first system is a mathematical engine and the second is an intelligent diagnostic pretest. The third component manages the knowledge which the program accumulates about the student and the fourth controls the tutoring activities themselves. The glue which holds the four systems together is the Structural Learning Theory of Joseph Scandura.

This paper describes the first phase of the construction of the program, the building of the mathematical engine and the intelligent pretest. The engine consists of rewrite rules culled from a structural analysis of the protocols of 210 students on a test comprising 40 linear equations. The rules may be activated as a set, to solve an equation, or one by one in any order to enable the student to observe their effect. The program has not yet failed to solve any well-formed linear equation put to it, and it will also simplify arbitrary linear expressions using the same set of rules. The structural analysis also enabled a hierarchy of equation types to be built such that an ability to solve an equation of one type implies the ability to solve equations lower in the hierarchy. The pretest exploits this to use the least number of questions to pinpoint the student's ability with respect to the equation types. The pretest is organised as a set of IF...THEN rules so that it would have general applicability.

EMMA: UN TUTEUR INTELLIGENT

EMMA est un tuteur informatisé d'algèbre conçu pour surmonter les trois principales faiblesses des logiciels éducatifs et pour être utilisé sur les micro-ordinateurs installés dans les écoles. EMMA est formé de quatre systèmes experts interreliés. Le premier système est un moteur mathématique et le deuxième, un prétest de diagnostic intelligent. Le troisième système gère les connaissances acquises par le programme au sujet de l'élève et le dernier assure le contrôle des activités de formation. Ces quatre éléments sont regroupés selon la théorie d'apprentissage structurel de Joseph Scandura.

Le document décrit la première phase de la constitution du programme, soit la construction du moteur mathématique et le prétest d'intelligence. Le moteur correspond à des règles de réécriture tirées d'une analyse structurale des protocoles de 210 élèves dans un test comprenant 40 équations linéaires. Les règles peuvent être appliquées toutes ensembles pour résoudre une équation, ou une à une, de façon à permettre à l'élève d'en observer l'effet. Jusqu'à présent, ce programme a permis de résoudre toutes les équations linéaires correctement formulées qui lui ont été soumises et il aura également pour effet de simplifier les expressions linéaires absolues au moyen du même jeu de règles. L'analyse structurelle a également servi à établir une hiérarchie des types d'équations de telle sorte que la capacité de résoudre une équation d'un type donné suppose que l'on peut résoudre toute équation de niveau inférieur dans la hiérarchie. Le prétest est basé sur cette prémisse et utilise le nombre le moins élevé possible d'équations pour déterminer la capacité de l'élève relativement aux types d'équations. Le prétest est structuré selon un jeu de règles de type SI...ALORS, ce qui lui assure des possibilités d'application dans un grand nombre de domaines.

EMMA: AN INTELLIGENT TUTOR

Martyn Quigley

Memorial University of Newfoundland
St. John's, NF A1B 3X8

1. Introduction

EMMA is a prototype computer tutor for linear equations written entirely in Lisp. It was designed to overcome the three main deficiencies of computer assisted learning materials identified by John Self, namely lack of knowledge of the subject matter to be taught, lack of knowledge about the student being taught and lack of knowledge about the process of teaching (Self 1974). In addition, Ronni Rosenberg has criticised intelligent tutoring systems for not being grounded in any substantive model of learning, nor preceded by protocol analysis (Rosenberg 1987). The design of EMMA avoids both these criticisms.

2. Research Method: The Structural Analysis

There is no empirically validated theory of learning, nor is there any psychological theory of what constitutes a useful physical representation of a mathematical concept (Resnick & Ford 1981 p249). The selection of an appropriate theory of learning about which to build an ITS is therefore a pragmatic decision to be based on the usefulness of the theory rather than its absolute "truth".

Children, when asked what they are doing (in mathematics) usually cite a rule of some kind, and typically only one rule is cited to justify any given response (Siegler & Richards 1979, Brainerd 1982 for example). There is a lot of evidence to suggest that children are very good at following algorithms and that almost all errors arise when they choose the wrong algorithm to follow or when they incorrectly sequence a correct algorithm (for example, Brown & Burton 1978 p 157). Indeed, this effect is so marked that it can be more fruitful to regard erring children as faithfully executing an incorrect algorithm than wrongly following a correct one (Young & O'Shea 1981, P154). This suggests the interpretation of mathematical ability as the holding of rules is strong enough to explain sufficient child behaviour to be a useful working model. The theory which Joseph Scandura constructs about this thesis, Structural Learning (Scandura 1973), forms the central basis of the design of EMMA.

Interpreting mathematical ability as rule-holding requires the rules which are held be identified. Since there is no guarantee that the rule which a student acquires is congruent to the rule which the teacher intended the student to acquire, it is not enough to determine a set of rules a priori from the mathematics. It is necessary to examine children's work and observe directly the rules which they use. Consequently, a structural analysis was carried out (Scandura 1986).

Sixty-two linear equations were selected (from several algebra texts) and analyzed to obtain a set of rules presumed to be correct. These rules are as follows.

Rule	Description	Example
RA	Add two numbers together	3x + 2 + 4 -> 3x + 6
RB	Multiplies two numbers together	3.4x -> 12x
RC	Divides out to obtain the solution	4x = 8 -> x = 2
RD	Move a number to the RHS	3x + 5 = 11 -> 3x = 11 - 5
RE	Commutative law of addition	2 + 3x -> 3x + 2
RF	Expands a set of brackets	3(2x + 5) -> 3.2x t 3.5
RG	Adds two x-terms	3x + 5x -> 8x
RH	Moves an x-term to the LHS	7x = 4x + 9 -> 7x - 4x = 9
RJ	Swaps the LHS and RHS	3 = 5x - 2 -> 5x - 2 = 3

For each of the 62 equations each solution step was identified with the appropriate rule name. This yielded, for each equation, at least one sequence of rules which gave a solution, and also gave, for each of these rules, a list of equations in which the rule occurred in the solution. For example, the equation -8 = x + 6 has at least two solution paths in terms of these rules.

```
   -8 = x + 6                -8 = x + 6
x + 6 = -8      RJ      -8 - x = 6        RH
  x = -8 - 6    RD      -x - 8 = 6        RE
  x = -14       RA        -x = 6 + 8      RD
                            -x = 14
                             x = 14       RC
```

The sequences which emerged from the solutions were used to partition the set of equations into what were termed Equation Types. For example:

```
Type 10          71 = 8 - 9x
Type 15     8x + 7 = 3x - 13
```

Twenty-eight types of equation were identified from the sample in this way, but six types were

excluded from further analysis since their solutions required multiple applications of similar sub-paths. For example, the equation

$$10 - 2(x + 5) + 4 = 3(5 + 2x) - 5(2x + 1)$$

requires the expansion of three sets of brackets.

A pair of equations of each of the remaining 22 types were selected and formed into a test of 44 items which was administered to a sample of 182 students varying in age from 12 to 15 and also of varying abilities.

The protocols returned were analyzed to determine which of the rules identified by the teachers were actually used, which other correct rules the students used and which incorrect rules existed (and the sequences and paths in which they were used). All of the (assumed correct) rules identified by the teachers were found in the protocols except for RJ, the rule which swaps the left hand side of an equation with the right hand side. A number of other correct rules were seen but all of them could be broken down into assemblages of the teacher identified rules. Several incorrect rules, termed malrules were seen, those which were each found in more than five scripts were:

Rule	Description
MRC	4x = 8 -> x = 2
	but 8x = 4 -> x = 2
MRD	3x + 5 = 11 -> 3x = 11 + 5
MRE	2 - 3x -> 3x - 2
MRF1	3(2x + 5) -> 6x + 5
MRF2	3(2x + 5) -> 6x + 8
MRF3	-3(2x - 5) -> -6x - 15
MRF4	3(2x + 5) -> 6x + 35
MRH	7x = 4x + 9 -> 7x + 4x = 9

A number of errors were bizarre. For example,

$$6x + 3 = 33 = 301 \quad \text{and} \quad 2 - 4x = -10 = 16$$

Whilst the second mistake may have been due to the child adding up all the figures in the equation, there is no obvious explanation for the first error. Solutions which contained such errors were excluded from the further analysis.

A split half reliability coefficient was calculated between the set of first pairs and the set of second pairs. The result, 0.84, is in rough agreement with prior laboratory and class experiments (Scandura & Durnin 1978, Durnin & Scandura 1973). In the actual construction of the tutor, eight correct rules were used together with 11 malrules and these rules were sufficient to account for 85% of the remaining protocols. The rest of the protocols contained mistakes which typically occurred only once.

The structural analysis enabled a hierarchy of equation types to be built which is an empirically verified partial ordering of the selected equations according to difficulty. This hierarchy, shown in Figure 1, is used to guide the instructional progression and also to sequence the questions in the pretest.

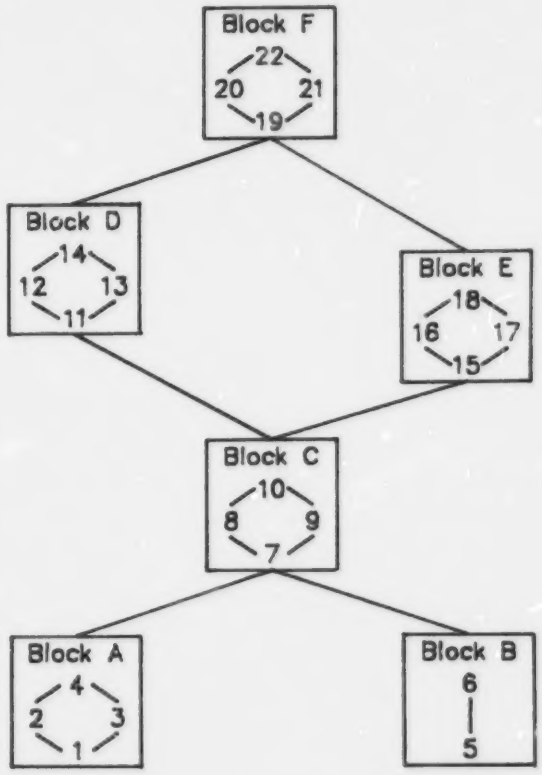


Fig. 1

3. Research Method: The Mathematical Engine

EMMA's mathematical engine is comprised of surgical IF-THEN rules which implement the rules described above and are similar to those used by Derek Sleeman in the Leeds Modelling System (Sleeman 1975). In essence, an equation is selected and each rule is invited to apply itself to that equation. Typically several rules would indicate they apply. EMMA decides which one of the several rules should act on the equation, permits the selected rule to rewrite the equation and then once again invite rules to apply themselves. This cycle of events repeats until no more rules apply in which case the equation is solved or the expression is fully simplified.

Each rule is coded as a frame-like structure (Minsky 1975) with each slot containing executable Lisp code. When a rule is invited to apply itself, a one-way unification pattern matcher (adapted from Winston & Horn, 1984) matches the rule's IF template against the current equation to see whether the conditions needed for the rule to act are satisfied. Each rule is assigned a priority. If several rules are applicable, the one with the highest priority fires. The precise values of the priorities are not important, merely their value relative to each other. In order,

highest priority first, the rules are: RB, RF, RE, RG, RJ, RA, RH, RC. This method of conflict resolution ensures EMMA solves equations correctly, but it does not guarantee the shortest path is found.

EMMA stores the current equation as a list, not in the standard mathematical notation. EMMA can generate equations, read them from a file or accept them from the learner. For example, the equation

$$2 + 3(4x - 5) = -9x + 8$$

is represented internally as

(NIL 2 + 3 * (4 * X + -5) = -9 * X + 8 NIL)

The NILS at either end are necessary because EMMA's matching procedure uses wildcards which can match successfully with one or more elements, but not with zero or more elements. It is easy to amend the matcher to correct for this, but the computational penalty is severe, an approximate doubling of processing time. Since the learner never sees the list form of the equation, elegance is sacrificed for efficiency in this case.

4. Research Method: The Pretest

When a student first comes to EMMA, the program has no knowledge about that learner and needs therefore to make an assessment of the learner's ability to solve equations. Traditionally, this would be done by a pretest in which the student is invited to solve equations beginning with the easiest and continuing through progressively harder equations until the student can do no more. The equation hierarchy derived from the structural analysis enables us to place a student more efficiently. We begin the student on an equation in the middle of the hierarchy, say Type 10. If the student can solve the equation then we can assume the student can solve equations lower in the hierarchy and there is no need therefore to ask the student to solve any of them. If the student cannot solve the initial equation then we can assume that it cannot solve any equations higher in the hierarchy and therefore we need ask none of them. We have partitioned the set of equation types into two sets, and placed the student into one of them. We repeat the process until the partition in which the student is placed contains only one equation type. This is the point at which the tutorial instruction should begin. Such a strategy in the traditional setting would require continual monitoring by the teacher.

The control of the process is done by two sets of IF-THEN rules organised as a two production systems. The first encodes the hierarchy in 23 rules like:

```
(IF (JIM CAN DO TYPE 7))
(THEN (JIM CAN DO TYPE 6))
```

A cycle of this system thus yields a list of the equation types which Jim can do. The second system contains 25 rules to control the flow of the test, for example

```
(IF (JIM HAS DONE TYPE 10)
(THEN (SET-NEW-QUESTION TYPE 15)
```

This form of rule was adopted because it permits great flexibility in the nature and scope of the knowledge which could be represented.

5. References

Brainerd C J (ed), 1982, Children's Logical and Mathematical Cognition, Springer-Verlag

Brown J S & Burton R R, 1978, Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, Cognitive Science 2: pp 155-192

Durnin J H & Scandura J M, 1973, An algorithmic approach to assessing behaviour potential: comparison with item forms and hierarchical analysis, Journal of Educational Psychology, 1973, 65, pp 262-273

Minsky M L, 1975, A Framework for Representing Knowledge in Winston P H (ed), 1975, The Psychology of Computer Vision, McGraw Hill pp 211-277

Resnick L B & Ford W W, 1981, The Psychology of Mathematics for Instruction, Erlbaum

Rosenberg R, 1987, A critical analysis of research on intelligent learning systems, Educational Technology 27 (1), November 1987

Scandura J M, 1973, Structural Learning, Gordon Breach

Scandura J M & Durnin J H, 1978, Assessing behaviour potential: adequacy of basic theoretical assumptions, Journal of Structural Learning, 1978, 6, pp 3-47

Scandura J M, 1986, An Intelligent Ruletutor CBI System for Diagnostic Testing and Instruction, Journal of Structural Learning, Vol 9, No 1, pp 15-61

Self J A, 1974, Student Models in Computer Assisted Instruction, International Journal of Man Machine Studies, 6: pp 261-276

Siegler R S & Richards D D, 1979, Development of Time, Speed and Distance Concepts, Developmental Psychology, 15: pp 228-298

Sleeman D, 1975, A Problem-Solving Monitor for a Deductive Reasoning Task, International Journal of Man-Machine Studies, Vol 7, No 2, pp 183-211

Winston P H & Horn B K P, 1984, LISP (2nd Edition), Addison-Wesley

Young R M & O'Shea T, 1981, Errors in Children's Subtraction, Cognitive Science 5: pp 153-177

AUTOMATIC ERROR-CORRECTION IN LEARNING LISP

R. Charron, F. Schmalhofer, G. Cartwright
Dept. of Educational Psychology & Counselling
McGill University
Montreal, PQ

Learning either a new programming language like LISP (F. Schmalhofer, O. Kühn, 1988) or application software (Carroll, Mack & Lewis, 1985) can be a very difficult and frustrating process. In an attempt to minimize these difficulties, a tutoring component was developed on an IBM PS/2 to unobtrusively monitor a user's interactions with the LISP system. When the tutoring component detects an error or inefficient interactions, it may intervene. The error corrector can then generate a correct input to the LISP system so that it is as similar to the erroneous user's input as possible while being correct. In addition, the tutoring component can provide active help when the user applies inefficient methods for achieving some goal.

To evaluate the system, eighteen subjects were randomly assigned to three different conditions. In the control condition, subjects learned by exploring without any help from the tutoring component. In the experimental conditions, assistance was either always provided or given selectively. The results of this study will be discussed in terms of guidelines for intelligent tutoring systems.

CORRECTION AUTOMATIQUE DES ERREURS PENDANT DANS L'APPRENTISSAGE DU LANGUAGE LISP

Il est bien connu que l'apprentissage d'un nouveau langage de programmation comme LISP (F. Schmalhofer, O. Kühn, 1988) ou d'un logiciel d'application (Carroll, Mack & Lewis, 1985) peut être très difficile et très frustrant. Pour tenter de minimiser ces difficultés, un tuteur a été conçu et implanté dans un IBM PS/2, pour contrôler, sans lui nuire, les interactions de l'utilisateur avec le system LISP. Lorsqu'il décèle une erreur ou des interactions inefficaces, il peut intervenir. Le correcteur d'erreurs peut alors générer une entrée que le système LISP peut accepter, mais une entrée qui soit aussi semblable que possible à la donnée erronée de l'utilisateur tout en étant valide. En outre, le tuteur peut fournir à l'utilisateur un soutien actif lorsque ce dernier utilise des méthodes inefficaces pour atteindre un but. Dans une expérience, on a tenté d'évaluer l'utilité de pareils outils dans un programme d'enseignement assisté par ordinateur.

Dix-huit personnes, qu'on a soumises, au hasard, à trois conditions différentes, ont participé à cette étude. Le premier groupe a appris par tâtonnement, sans l'aide du tuteur. Au deuxième, le logiciel a offert une aide constante alors que dans le cas du troisième, cette aide était occasionnelle. On étudiera les résultats de cette étude en vue de déterminer si l'on peut établir, à partir de cette dernière, des lignes directrices pour l'établissement de tuteurs intelligents.

AUTOMATIC ERROR-CORRECTION IN LEARNING LISP

R. Charron, F. Schmalhofer & G. Cartwright

Department of Educational Psychology & Counselling
McGill University
Montreal, Quebec

Educational and cognitive research is concerned with the acquisition and application of procedural knowledge in performance situations. With the increasingly widespread use of sophisticated computer software systems in education, business, and government, producing instruction and performance support for students and personnel who use these systems is emerging as a major issue in education and training. Often, a common failing with new applications software packages and programming languages is that the learners are provided with inadequate tutorials or supporting documentation.

Intelligent Tutoring Systems - Tutoring versus Discovery Learning

Intelligent tutoring systems (ITS) attempt to provide a learner with the same instructional opportunity as provided by a private human tutor, that is, effective and efficient learning strategies (Anderson, Boyle, & Reiser, 1985). As with a private human tutor, a difficulty with ITS in the teaching of problem-solving is to maintain a fine equilibrium between tutoring (guided practice) and discovery learning (unguided practice). Dede (1986) views an intelligent tutor's goal as transmitting descriptive knowledge to enhance the understanding of procedures rather than teaching process skills directly. The intelligent coach provides a model of expert problem-solving which the student can access while practicing process skills in a realistic context. One feature of the tutor is that the learner's understanding of a subject can first be compared to an expert's knowledge and then any lack of knowledge can be supplemented. Kimball (1982) views the tutor as stimulating learners to produce better solutions resulting in better judgment. On the other hand, stimulating learners to produce better solutions resulting in better judgment. On the other hand, discovery learning is a distinct process in which the learners are exploring and learning on their own with no explicit idea of student goals, performance, or state, and no built-in capacities to direct them toward some purposeful behaviour (Becker, 1987; Kimball, 1982). The lack of tutoring may decrease motivation and inhibit progress in developing overt strategies, while too much tutoring may prevent learners from understanding the problem structure (Brown & Burton, 1982; Dede, 1986). The answers are provided but not the method nor the comprehension of the solution. Tutoring gives students additional information and working memory support (information the student is likely to forget) in order to transform errors into constructive exploration, learning, and problem-solving (Anderson et al., 1985; Brown & Burton, 1982).

The controversial issue of whether the learner should be interrupted and corrected immediately by a tutoring system or should be allowed to learn by discovery persists in current studies. In what way will different degrees of assistance affect learners' programming knowledge acquisition? This study evaluated the effectiveness of no assistance, immediate assistance, and intermittent assistance, on performance of programming tasks.

Intelligent Tutoring Systems

The literature on ITS suggests that learning and retention of new information are heightened when certain instructional strategies are presented in interesting, motivating, and procedural ways (Givner, 1986; Schmalhofer, 1987). Carroll, Mack, and Lewis (1985) found that fewer details and less rigid instructions made remembering easier for the learners and allowed for learning through exploration. Other researchers have found that learning by exploration is more effective when the learners had prior computer experience (Schmalhofer & KÜhn, 1988).

In order for computer learners to attain the goal of skill acquisition and enhance their general problem solving ability, ITS should understand **what**, to **whom**, and **how** they are teaching and therefore should tailor the help content and the method to the learners' needs (Dede, 1986; Ohlsson, 1987). It should monitor unobtrusively, detect errors or limited strategies, and inform the learner with relevant comments, precise feedback, and as little time delay as possible so as not to impede the learning process (Anderson et al., 1985; Brown & Burton, 1982; Bumbaca, 1986; Dede, 1986; Fischer, Lemke, & Schwab, 1985). The timing of the tutor's intervention should be viewed as an essential component of ITS teaching strategies (Ohlsson, 1987). Moreover, learners' working memory load may be facilitated by having the tutor information encoded on the screen with immediate feedback about their errors (Anderson et al., 1985).

The LISP tutor study (Anderson et al., 1985) comprised an ideal model of LISP programming, an interface for communicating the goal of the programming tasks, instructions, and immediate feedback. Two studies were reported using this LISP tutor. In the first, no significant difference was found between the two tutored conditions but both were significantly faster than the no assistance group. The second study showed that the computer-tutored students spent less time completing the problems and scored better on the final exam than the students who worked independently.

Lewis and Anderson (1985) used dungeon quest-like games to determine whether it was better to have learners discover errors or to inform them immediately after an error had been made. It was determined that the students who received immediate feedback performed better when the number of correct moves was used as the criterion. Anderson (1987) noted that the type of feedback was critical in determining effectiveness. He stressed that feedback need not give the correct answer but rather alert the learner that an error had been made. It was also discovered that when subjects were allowed to learn by exploration the learners were better able to recognize their errors (Lewis & Anderson, 1985). The authors suggested that if students wanted to learn a skill such as debugging computer programming languages, then it was more beneficial for them to make errors in their programs so that they would have something to debug.

Reiser et al., (1988) developed a model tracing tutor called Graphical Instruction in LISP (GIL) which provided problem-solving instruction by monitoring a student's solution, comparing each step to its own domain knowledge, and providing feedback when the student needed guidance or demonstrated a misconception. The tutor traced the learners' reasoning as each part of a solution of a LISP example was entered into the system. When a correct path was taken the tutor was silent and followed the path; when a bug or error was detected the tutor interrupted with a positive statement about the correct steps taken by the student and then pointed out where a mistake had been made and how it could be improved. A built-in help facility provided a hint to allow the student to continue. Since the tutor intervened immediately there was no delay and no discovery learning was possible.

Some major problems associated with tutoring systems are that: (a) the tutor does not know what cues should be used to determine when to interrupt students' problem-solving activities; (b) the tutor does not know what to say when it does interrupt; and (c) the tutor does not know what concepts the students already knows or, perhaps more importantly, what misconceptions they hold (Brown & Burton, 1982; Linn & Dalbey, 1985; Ohlsson, 1987; Peelle, 1983). Dede (1986) feels that three fundamental issues for tutors are whether to intervene during the learners' entering of information, which presentation strategy to use, and how much to say. In the exploration environment, intervention results in interruption by the tutor when systematic errors are detected. In the tutoring environment, intervention involves determining either when to impart new information or to debug the learners' pattern (Dede, 1986). Too frequent interventions and discontinuous information flow diminish students initiative and decrease motivation to learn (Brown & Burton, 1982). Immediate feedback is thought to deprive learners of opportunities to discover by themselves (Brown & Burton, 1982). Discovery learning is considered to develop cognitive skills that would allow learners to detect and correct their own errors.

According to Linn and Dalbey (1985) the challenge of cognitive educational research is to design instruction that will allow students to recognize and correct their faulty approaches to problem-solving.

Becker (1987) suggests that for some learners the tutor or "intelligent instructor" guides, facilitates, and structures the learning process. The work by both Brown and Burton (1982), and Dede (1986) stress that the information provided by the intervention should be directed at a particular weakness, appear useful in the immediate situation, and be demonstrably superior to the learner's misconception.

Considering the previous studies and their results, the issue of whether immediate help and feedback or discovery learning was more beneficial, effective, and efficient to the learning process is intriguing. The rationale behind the design of the present study was to be able to determine the most effective way to assist learners. Should learners mistakes be corrected always, selectively, or never?

Method

Eighteen students with varying levels of computer experience and no prior exposure to LISP volunteered to participate in this research study (11 males, 7 females, aged 18-40 with a mean age of 24). The students were classified according to six levels of computer programming experience by the number of lines they had previously programmed. The students were then randomly assigned to three treatment conditions which provided different degrees of feedback. The conditions were:

1) Immediate Assistance: every mistake by the learner was corrected immediately and feedback presented.

2) Selecti[illegible] Assistance: assistance was provided only after two errors occurred in succession or after a learner engaged in two non-optimal learning sequences. Some delay was allowed permitting guided discovery learning.

3) No Assistance: no help or cues were provided allowing learning by unguided discovery.

In the two assistance conditions, the tutor responded with the following feedback "a possible and correct input is . . . " until a correct response was entered by the student.

The experiment comprised three phases:

A. Reading of the text: all subjects were given an introduction to the experiment on screen. Following this, there was an explanation on the LISP programming language and descriptions of the LISP functions.

B. Exploration condition: in this phase, the subjects explored at will and learned about the four LISP functions - first, rest, set, and list. In order to aid the learners coloured parentheses indicated the different levels of nesting in the LISP expressions. A spelling program checked for simple reversals or mistyping of the LISP functions.

C. Programming and Evaluation: in this phase, subjects were presented with various programming and evaluation tasks requiring them to generate an input to the LISP system.

Results and Discussion

The data were treated by 3 x 6 analysis of variance (ANOVA) design (treatment conditions x number of previously programmed lines).

A central finding was that tutoring strongly influenced the amount of time learners spent in completing both programming and evaluation tasks. Analysis of programming time showed that both the tutored groups took significantly less time than the no assistance group ($F(2,15)=5.30$, $p<.05$). Analysis of evaluation time suggested that the tutored groups took significantly less time in completing evaluation tasks than the no assistance group ($F(2,15)=12.12$, $p<.01$). In fact, though the difference in time between the two tutoring conditions was not significant, subjects in the tutoring conditions took significantly less time to complete the assigned problems of programming and evaluation tasks. This corroborates the findings of Anderson et al., (1985).

Another finding concerned the amount of computer experience related to the number of correct results. There was little variability among the six levels of programming experience. When these were pooled to form high and low programming experience groups, a significant difference was found between the number of correct responses on the evaluation tasks ($F(1,16)=4.90$, $p<.05$). Those with more programing experience completed more evaluation problems correctly. This supports the findings of both Linn and Dalbey (1985), and Schmalhofer and KÜhn (1988) who found that those learners with experience in several programming languages were more likely to develop problem-solving skills as well as robust, general templates for programming.

In the midst of the exploration phase, many subjects who had learned LOGO felt that this would help them in learning LISP functions. In fact there was no evidence of this.

Subjects who received tutoring thought that the tutor had helped them by giving them the correct responses after they had made errors. Yet when the tutor interrupted the subjects, they often became confused. In order to understand their erroneous LISP input they spent considerable time rereading the statement and wondering why the tutor's message appeared. This seemed to keep them on task. Several subjects commented that the tutor's responses gave them good direction; they floundered less once they comprehended the proper LISP functions.

ITS is acknowledged as being generally effective but further research needs to determine the specific nature of its effectiveness (Sleeman & Brown, 1982). Perhaps one of ITS's potentials is to serve as a research tool in the enhancement of human learning and to provide a new research environment for the study of cognition.

Acknowledgements

We are grateful to Caroline Kato and Carolyn Klein for their continuing support and assistance with this paper. This research was supported by an equipment grant to McGill University from IBM Canada.

References

Anderson, J. (1987). Skill acquisition: Compilation of weak-method problem solutions. Psychological Review, 94, 192-210.

Anderson, J., Boyle, C., & Reiser, B. (1985). Intelligent tutoring systems. Science, 228, 456-462.

Becker, L. (1987). A framework for intelligent instructional systems: An artificial intelligence machine learning approach. Journal of the Association for Educational and Technology, 24, 128-136.

Brown, J., & Burton, R. (1982). An investigation of computer coaching for informal learning activities. In Sleeman, D., & Brown, J.S. (Eds.), Intelligent tutoring systems. New York: Academic Press.

Bumbaca, F. (1986). An experimental intelligent computer-based coach for teaching logic. Proceedings of the Fifth Canadian Symposium on Instructional Technology (pp.145-150). Ottawa, ON: National Research Council.

Carroll, J., Mack, R., & Lewis, C. (1985). Exploring a word processor. Human-Computer Interaction, 1, 283-307.

Charron, R. (in preparation). The influence of different degrees of assistance in intelligent tutoring systems. Montreal: McGill University.

Dede, C. (1986). A review and synthesis of recent research in intelligent computer-assisted instruction. International Journal of Man-Machine Studies, 24, 329-353.

Fischer, G., Lemke, A. & Schwab, T. (1985) Knowledge-based help systems. In Borman, L. & Curtis, B. (Eds.), Human Factors in Computing Systems. San Francisco: 161-167.

Givner, B. (1986). Implementing computers in the classroom. Education Research Quarterly, 10, 25-29.

Kimball, R. (1982). A self-improving tutor for symbolic integration. In Sleeman, D., & Brown, J.S. (Eds.), Intelligent tutoring systems. New York: Academic Press.

Lewis, C. & Anderson, J. (1985) Discrimination of operator schemata in problem solving: Learning from examples. Cognitive Psychology, 17, 26-65.

Linn, M. & Dalbey, J. (1985). Cognitive Consequences of programming instruction: Instruction, access and ability. Educational Psychologist, 20, 191-206.

Ohlsson, S. (1987). Some principles of intelligent tutoring. In R. L. Lawler & M. Yazdani (Eds.), Artificial intelligence and education: Learning environments and tutoring systems, 1, (pp. 203-237). Norwood, NJ: Ablex.

Peelle, H. (1983). Computer metaphors: Approaches to computer literacy for educators. Computer Education, 7, 91-99.

Reiser, B., Friedmann, P., Gevins, J., Kimberg, D., Ranney, M., & Romero, A. (1988). A graphical programming language interface for an intelligent LISP tutor. Proceedings of the Conference on Human Factors in Computing Systems (pp. 39-43). Washington, DC: Addison-Wesley.

Schmalhofer, F. (1987). Mental models and procedural elements approaches as guidelines for designing word processing instructions. Human-Computer Interaction - Interact'87, (pp. 269-274).

Schmalhofer, F. & KÜhn, O. (1988). Acquiring computer skills by exploring versus demonstration. Proceedings of the Cognitive Sciences Society (pp. 724-730). Hillsdale, NJ: Lawrence Erlbaum Associates.

Sleeman, D. & Brown, J.S. 1982. Intelligent Tutoring Systems. New York: Academic Press.

COURSE DEVELOPMENT WITHIN A SOPHISTICATED COMPUTER-MANAGED LEARNING ENVIRONMENT

M. Jones, K. Wipond
Advanced Technologies Department
Alberta Research Council
Calgary, AB

In recent years there has been a growing trend in North America towards competency-based education (often termed "mastery learning"). Because competency-based education advocates individually-paced learning there has been a corresponding increased demand for sophisticated Computer-Managed Learning Systems and computer-based curricula. The "Expert" Computer-Managed Learning Environment (CML) has been designed to address these needs. It includes four main phases: 1) an environment for curriculum development; 2) an environment for course development from curricula; 3) a component that guides, monitors and evaluates students as they progress through courses developed in phase 2; and 4) an expert system for curriculum and course evaluation.

In this paper we briefly describe the overall project and concentrate on the second phase. Phase 1 development results in a rich on-line representation of curricula. In phase 2 the developer is provided with a set of course development tools, with which he/she can effectively build up an on-line representation of a course, based on phase 1 curricula. The development process includes the selection of course components, specification of ordering and prerequisite constraints, the display of the resulting course map, as well as the specification and display of other course features such as schedules and marking schemes. This representation of the course facilitates individual student routing and individually paced student progress. Phase 2 of CML has been written in C to run on the microvax.

CONCEPTION DE COURS DANS UN CADRE PERFECTIONNÉ D'ENSEIGNEMENT AUTOMATISÉ

Depuis quelques années, il y a en Amérique du Nord une tendance croissante vers le recours à la formation basée sur les performances attendues (souvent appelée pédagogie de la maîtrise). Puisque la formation basée sur les performances attendues préconise l'apprentissage auto-rythmé, on a vu une augmentation correspondante de la demande pour des systèmes perfectionnés d'enseignement automatisé et des programmes d'études informatisés. Le système d'enseignement automatisé "Expert" a été conçu pour répondre à cette demande. Il compte quatre grandes phase : 1) un cadre pour la conception de programmes d'études; 2) un cadre pour la conception de cours à partir des programmes d'études; 3) une partie qui guide, supervise et évalue les étudiants au fur et à mesure qu'ils progressent dans les cours établis à la phase 2; 4) une système expert pour l'évaluation des programmes d'études et des cours.

Dans cet exposé, nous décrivons brièvement le projet et nous insistons sur la phase 2. La réalisation de la phase 1 a donné pour résultat une large représentation en direct de programmes d'études. A la phase 2, le concepteur se voit remettre des outils de conception de cours avec lesquels il peut monter de façon efficace une représentation en direct d'un cours, d'après les programmes d'études établis à la phase 1. Le processus de conception prévoit le choix des parties du cours, la spécification de l'ordonnancement et des conditions préalables, l'affichage du plan du cours, ainsi que la spécification et l'affichage d'autres caractéristiques telles que les calendriers et les méthodes de marquage. Cette organisation du cours facilite l'orientation de l'étudiant et son cheminement auto-rythmé. On a écrit la phase 2 du programme d'enseignement automatisé en C pour pouvoir l'utiliser sur le Microvax.

COURSE DEVELOPMENT WITHIN A SOPHISTICATED COMPUTER-MANAGED LEARNING ENVIRONMENT

Marlene Jones and Kevin Wipond

Advanced Technologies Department
Alberta Research Council, Calgary, ALTA

INTRODUCTION

Although Computer-Managed Learning (CML) systems have been commercially available for many years, there is a growing demand for more sophisticated systems. This demand is partially fueled by the push towards *competency-based* education (also termed *mastery learning*). Competency-based education advocates curricula based upon explicitly stated expected performance outcomes by which a student's achievements can be measured. Hence, there is a corresponding increased demand for individually-paced instruction and computer-based curricula. In addition to this appeal for more sophisticated assessment and monitoring of a student's progress, there is also a need to provide assistance in the development of the various on-line curricula and courses. This is particularly significant in light of the fact that curricula are often developed by instructors who may have little background in the areas of curriculum development and instructional design.

The "Expert" CML environment is a large research initiative between Computer-Based Training Systems Ltd. (CBTS) and the Alberta Research Council (ARC) which was launched in the fall of 1986. The system's purposes include assisting developers in producing on-line curricula and courses, and monitoring individual student's progress according to the learning objectives specified within the curricula. The four main system components, termed *phases*, are as follows:

- phase 1 - an environment for curriculum development including expert advice and assistance
- phase 2 - an environment for course development from curricula including expert advice and assistance
- phase 3 - a component that guides, monitors and evaluates students as they progress through courses developed in phase 2
- phase 4 - an expert system for curriculum and course evaluation.

Within the first environment (phase 1) the user develops curricula by building up representations of all curricular content from the broad goals of a program of study through to the specifics of particular modules and submodules. Within the environment for course development (phase 2), a user can select and adapt curricula representations to meet particular requirements, specifying the preferred order of the material, dates for exams, mastery requirements, module study guides, etc. The tools and representation schemes of phases 1 and 2 allow the user to build up a rich on-line representation of one or more courses. In phase 3, students are enrolled in the courses and are guided by the system through the course material. The system routes the student through the course material based upon the student's performance; this may include routing the student to remedial or enrichment material, based upon the information stored within both phases 1 and 2 of the system. In phase 4, the tracking of student performance allows the system to suggest how the curricula and courses might be enhanced or changed to facilitate the learning process.

An important aspect of both phases 1 and 2 is the expert system monitor. The monitor mechanism oversees the user's curriculum and course development activities, providing advice which guides, and expertise which embodies principles of "good" design. Each time the user enters or modifies curricular information, the system assesses the action, and determines if it violates one of the expertise rules. If so, the system typically interrupts the development process to inform the user of the violation. (Advice rules do not interrupt the user). The user always has the option of ignoring any advice or expertise presented by the system. In fact, the user has the option of "turning-off" any expertise rule and may even turn off the expertise mechanism altogether (leaving the advice mechanism still functioning). In this way, the system is appropriate for users with a wide range of backgrounds and expertise. For a detailed description of the monitor mechanism, the reader should consult [4].

"Expert" CML can be accurately classified as an expert or knowledge-based system, as it captures a variety of expertise concerning curriculum and course development. Both the development of the system and its structure are representative of traditional expert system development. System development has been an iterative process

consisting of an interweaving of the following stages: knowledge acquisition, conceptual design, prototype development, and implementation. For a discussion of knowledge acquisition process, the reader should consult [2].

In this paper, we present a description of phase 2, the course development component. We begin with a brief overview of phase 1, the curriculum development component. For a more detailed description of phase 1, including the curricular representation schemes, the tools by which the user undertakes the development, the expertise and advice, the reader should consult [3,4]. A general overview of the system is contained in [1].

CURRICULUM DEVELOPMENT ENVIRONMENT

Phase 1 provides a rich curriculum representation, a set of tools to facilitate the curriculum development process, and a monitor by which to oversee the development process. The curriculum representation includes each of the entitities that make up a curriculum such as departments, programs, courses, topics, subtopics, modules, learning objectives, resources, evaluation items, learning activities, and taxonomies. Each of these entities has many attributes. For instance, among other things, courses have names, descriptions, goals, prerequisites, and taxonomies (by which learning objectives are categorized).

The basic objects that make up a curriculum are linked together in a default curriculum framework capturing the relationships that exist between them (Figure 1). For instance, courses have topics, and modules have objectives. When the user initiates the creation of a new course curriculum, she is given the option of using the default curriculum framework or modifying it to more aptly suit the needs of the particular course.

After specification of the curriculum structure, the user proceeds with the development of the on-line curriculum, specifying topics, subtopics, modules etc. The approach to curriculum development that the system *encourages* is one of top-down refinement where the curriculum is refined successively from broad goals of the course through to topics, subtopics, modules and possible submodules. Modules (and submodules) are refined further to objectives. Associated with the objectives are learning activities to aid a student in mastering the objective, and evaluation items by which a student's achievements can be assessed. In addition to the refinement relationships, prerequisite and corequisite relationships are also represented.

COURSE DEVELOPMENT ENVIRONMENT

In phase 2, the term *course* is used to mean the delivery of selected pieces of one or more curricula to a group of students during a particular period of time along with evaluation of those students. Typically, a phase 2 course is offered during a given semester to one or more classes of students. The system provides a set of tools and guidance for building an on-line representation of a course.

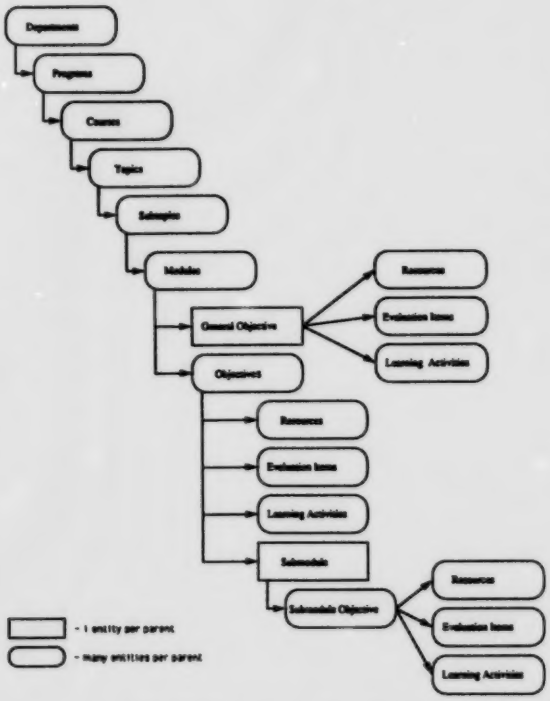


Figure 1. The Default Curriculum Framework

A phase 2 course consists of portions of one or more phase 1 curricula (such as particular topics, subtopics, modules and the associated objectives, learning activities and evaluation items). The instructor can specify an ordering of these curricular objects (in addition to the prerequisite and corequisite information already specified within phase 1). This ordering information is ultimately employed by the system (in phase 3) to determine the order in which individual students can progress through the course content. A graphical representation of this ordering information, the course map, is also produced by the system. A phase 2 course also typically includes the specification of assessments (assignments, tests and exams), possibly with dates associated with each. Relevant assessments and associated dates can be listed in the course schedule.

The stages of phase 2 course development are: the selection of pieces of phase 1 curricula to be included in the course, the specification of ordering information, viewing the course map, specifying the schedule, specifying the marking scheme, specifying module study guides and specifying other course-related information. The selection, ordering and viewing stages of phase 2 are usually interwoven in the course development process.

Typically a user will select pieces of the curriculum to be incorporated, specify some of the ordering information regarding these selected objects, view the resulting course map, select additional objects, add or delete ordering information, view the map again, until the map is deemed a reasonable representation of the material to be included in the course.

Assistance in the course development process is provided to the user in the form of help, advice and expertise. Help consists of explanations, including examples where appropriate, regarding each command and representation field. Advice reminds the user as to which information should be specified. Expertise captures some basic principles of course development.

The output of phase 2 is a specification of the course as it is to be implemented in phase 3. When routing a student through the course material, the system employs the ordering and scheduling information (as specified in phase 2) to determine what the student can and should do next, taking into account upcoming assessments for which the student may need to cover particular material or review. The individual student's performance will also determine the need for review or remediation.

THE SELECTION PROCESS

In the *selection process* the user selects material, from potentially any of the on-line curricula, to be included in a particular course offering. The user moves through the existing curricula selecting and de-selecting material such as topics, subtopics and modules. The selection of a topic generally indicates that all modules within that topic are to be included within the course (unless otherwise specified).

An interface is provided to facilitate this selection process. The user begins at the top of the curricula hierarchy with the list of departments, and then moves down through the curricula levels from department to module. At any particular level of the curricula the user sees a description of the current object (e.g. topic), a list of its goals, a list of the objects at the next level of the curriculum (e.g. subtopics), and a list of the prerequisites and corequisites for that object. All of this information is derived from the phase 1 curricula representation. The user can select the object in question to be included in the developing course. The selection of an object automatically causes the selection of all objects below that object. For instance, the selection of a topic causes the automatic selection of all of its subtopics and their modules, etc. The user is, of course, able to de-select any of these objects.

ORDERING THE SELECTED MATERIAL

Once the user has selected the curricular material, she must decide how it should be ordered for presentation to the student. Typically, prerequisite and corequisite relationships specified within phase 1 will hold between some of the selected material and these requisites are constraints on the order that the material will be presented to the students. The user generally provides additional constraints regarding the order in which students can progress through the material. These further constraints are different from requisites and usually reflect an instructor's preferences as to the order of progression through the material.

The ordering interface consists of several screens containing the list of selected course material, a list of operators that can be used when building up an ordering specification, a list of the current ordering specifications, a list of any prerequisite or corequisite relationships that hold between the selected material, and a statement of the current ordering specification (if any). Within this ordering interface a number of fundamentally different things can be accomplished. Ordering specifications, which are statements about constraints on the ordering of the course material, can be built up. As part of this process, conceptual groups consisting of selected material can be created and can then be ordered themselves. New prerequisite and corequisite relationships can be specified which will be reflected within the phase 1 curriculum representation. Finally, existing ordering specifications can be deleted.

There are three types of conceptual groups: AND groups, OR groups, and XOR groups. In the simplest case, the student must complete all elements of an AND group, at least one element of an OR group, and exactly one element of an XOR group. This simple case is further complicated by ordering specifications made between elements within a group. The user builds up ordering specifications by alternately selecting an object (curricular material or group) followed by an operator (AND, OR, XOR, ->, END). The -> operator is read "must be completed before". The END operator signifies the completion of an ordering specification. For example, an ordering specification such as:

module_a AND module_b AND group_G ->
module_c AND group_G2

means that module_a, module_b and group_G must all be completed before starting module_c and before starting group_G2.

As each ordering constraint is specified by the user, it is checked for consistency with the other specified ordering information. The system detects contradictions where one ordering specification implies a certain order holds between two objects and a later specification implies a conflicting order.

VIEWING THE COURSE MAP

The course map is a graphical representation of the ordering information and is meant to aid the user in the ordering process. It displays the constraints

on the order in which students can progress through the material.

Curricula is multi-layered in structure with topics refined into subtopics, and subtopics into modules. In addition the user can impose further layering by combining objects together into conceptual groups. Objects at any level of this structure can be ordered with other objects at other levels. So for instance, a topic could be a prerequisite for a module while a group could be ordered before some subtopic. The course map interface reflects this multi-layered structure by allowing the user to view ordering constraints between objects at any particular level. So for instance the user can view the topic level of the course map which is a representation of the topics in the course and any ordering information that holds between them or pertains to them. Similarly, each layer of the curriculum structure has its own level map.

At each level, the graphical representation includes a display of the objects within the course map at that level and the ordering relationships that pertain to these objects. The objects are placed left to right with those on the far left having no objects ordered previous to them. These are the objects that can be started as soon as a student enrolls in the course. Other objects are placed so that lines can be drawn between them and the objects that are ordered before them. In this way the ordering relationships between objects are reflected by the lines drawn between them. Within the course map there are different types of arcs depending on the type of order relationship that holds between the objects (OR, AND, XOR).

OTHER COURSE DEVELOPMENT ACTIVITIES

Several other phase 2 interfaces allow the user to specify further information pertaining to the course being developed. In the course schedule interface the user associates dates with the various activities in the course such as completion of topics, modules, exams and assignments. In addition, the user can indicate which elements of the course are optional. A marking scheme for the course indicates which assignments, exams and labs will contribute to the student's mark and the weighting factor for each. Module study guides are printed information to be presented to the student to aid progress through the module. This information typically includes learning objectives, learning activities and practice evaluation items. The developer is able to use system defined templates for module study guides, adapt these to meet individual needs or create new templates. Finally, there is an interface which allows the user to specify other course related information including details about the various instructors for the course, the location for the course, when pretesting is allowed, when the student is allowed to challenge course material, etc.

CONCLUSIONS

The "Expert" CML system is a sophisticated computer-managed learning environment designed to assist instructors with the development of on-line curricula and courses, as well as guide and monitor students' progress through the various courses. The course development environment (phase 2) is one of the key components of the system, allowing the user to develop courses based upon the on-line curricula developed within phase 1. Assistance with the course development process is provided via the expert system monitor.

At the time of writing, implementation of the curriculum development environment (phase 1) is nearing completion and a preliminary version of the course development environment (phase 2) has been prototyped using Lisp and C on a MicroVax. The design was then enhanced and the implementation of the delivery system is now underway. CBTS expects to complete the implementation of this component during the summer of '89, and launch the implementation of phases 3 and 4 later in '89.

REFERENCES

1. Jones, M., & Massey-Hicks, M., (1987) "Expert CML: The Next Generation", *Proceedings of the Computer-Assisted Learning in Tertiary Education (CALITE) Conference,* Sydney, Australia, invited paper.

2. Jones, M., & Wipond, K., (1989) "Intelligent Environments for Curriculum and Course Development", in: P. Goodyear (ed), *Teaching Knowledge and Intelligent Tutoring,* Ablex Publishing, (to appear).

3. Wipond, K., & Jones, M. (1988). "Curriculum and Knowledge Representation in a Knowledge-based System for Curriculum Development", *Proceedings of ITS'88: International Conference on Intelligent Tutoring Systems,* Montreal, p97.

4. Wipond K., & Jones, M. (1988). "Monitoring the User Within a Sophisticated Computer-Managed Learning Environment", *Proceedings of IASTED'88: Expert Systems Theory and Their Application,* Los Angeles, 14-18.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of fellow team members Julia Driver, Doug Konkin, Sandy Paterson, Chris Stang, Rick Steele, Alec Stephen and Dave Stephen, who have and continue to play critical roles in ensuring the success of the "Expert" CML project.

UN SYSTÈME TUTORIEL INTELLIGENT: MODÈLE THÉORIQUE ET RÉALISATION

G. Gauthier
Dép. de mathématiques et d'informatique
Université du Québec à Montréal
Montréal, PQ

G. Imbeau
Dép. d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, PQ

Les systèmes tutoriels intelligents constituent une application de l'intelligence artificielle à l'éducation. Dans cette communication nous présentons premièrement un modèle théorique pour les systèmes tutoriels. Le modèle incorpore les concepts de curriculum, de modélisation de l'étudiant, de syllabus de cours et d'enseignement. On associe un automate fini déterministe à un système tutoriel. Ensuite, dans le contexte du projet Héron, nous décrivons comment certaines composantes du modèle ont été réalisées dans notre prototype d'un système tutoriel intelligent pour l'enseignement du traitement de texte.

Le projet Héron, sous la direction du professeur Claude Frasson de l'Université de Montréal, a parmi ses objectifs celui de réaliser des systèmes d'enseignement assisté par systèmes experts dans le domaine de la formation industrielle. Des prototypes sont développés en utilisant la coquille ART de Inference. Le projet est à la fois interdisciplinaire (regroupant des chercheurs en technologie éducative, sciences de l'éducation et informatique) et inter-universitaire.

AN INTELLIGENT TUTORING SYSTEM: THEORETICAL MODEL AND IMPLEMENTATION

Intelligent tutoring systems are an application of artificial intelligence to education. In this paper, we will first present a theoretical model for tutoring systems. The model will involve the concepts of curriculum, student profile, course syllabus and teaching. A deterministic finite automaton will be associated with a tutoring system. Then, in the context of the Héron project, we will describe how certain components of the model have been implemented in our prototype of an intelligent tutoring system for teaching word processing.

The objectives of the Héron project, directed by Professor Claude Frasson of the University of Montreal, include the implementation of teaching systems assisted by expert systems in the field of industrial training. Prototypes are being developed using the ART inference shell. The project is both interdisciplinary (bringing together researchers in instructional technology, education and data processing) and interuniversity in nature.

UN SYSTÈME TUTORIEL INTELLIGENT: MODÈLE THÉORIQUE ET RÉALISATION

Gilles Gauthier, Gilles Imbeau

Université du Québec à Montréal,
Université de Montréal

INTRODUCTION

Les travaux de recherches présentés ici ont été réalisés dans le cadre du projet HÉRON. Ce projet est à la fois interdisciplinaire (regroupant des chercheurs en technologie éducative, en sciences de l'éducation et en informatique) et interuniversitaire. Le groupe s'intéresse aux systèmes tutoriels intelligents reliés au domaine de la formation. Pour une description des différents axes de recherche du projet HÉRON, voir [4].

Certains aspects distinguent un **s**ystème **t**utoriel **i**ntelligent (STI) d'un système d'**e**nseignement **a**ssisté par **o**rdinateur (EAO) classique: le système connaît la matière à enseigner (il peut résoudre les problèmes qu'il demande à l'étudiant), il possède un représentation de ce que l'étudiant connaît ou ignore de la matière et de ses incompréhensions, et enfin il connaît des stratégies pédagogiques et sait en faire bon usage.

Connaissance du domaine

Anderson [1] réfère à cette *connaissance du domaine* comme étant l'une des deux sources d'intelligence d'un STI; l'autre réside dans les principes qui guident le choix de stratégies pédagogiques et les méthodes de mise en oeuvre de celles-ci. Dans la plupart des STI, la connaissances du domaine est réalisée conjointement par une **base de connaissances du domaine** et par un module **expert du domaine** qui en tire les conséquences.

Connaissance de l'étudiant.

Le système doit être en mesure de déduire approximativement ce que l'étudiant connaît de la matière. Le composant *connaissance de l'étudiant* est réalisé conjointement par une base de données appelée **modèle de l'étudiant** et un module de **diagnostic** qui analyse les performances de l'étudiant et met à jour le modèle de l'étudiant. Les réactions, les erreurs et les intérêts de l'étudiant sont pris en compte pour qu'il bénéficie d'un cheminement individualisé et d'un soutien pédagogique adapté à ses besoins [7].

Expertise pédagogique

Un STI doit posséder une *expertise pédagogique*, c'est-à-dire des stratégies d'enseignement pour réduire la différence entre les performances de l'étudiant et celles d'un expert humain. Dans un STI, le composant chargé de l'enseignement proprement dit est appelé **expert pédagogue**. Il couvre deux aspects: la **planification** et la **supervision**. Les théories en pédagogie [2] et en enseignement assisté par ordinateur [9] peuvent ici être mises à contribution pour implémenter ce composant, ce qui, lors du développement d'un STI, force la formalisation des processus d'apprentissages, lesquels sont souvent intuitifs. Un STI peut s'avérer un instrument d'expérience très intéressant pour les pédagogues puisqu'il s'agit d'un système qui peut être analysé et critiqué par des experts dans le but d'améliorer les techniques d'enseignement qu'ils utilisent.

Interface étudiant-système

La communication entre l'étudiant et le système tutoriel est réalisée à l'aide d'un **micro-monde** [6, 8]. Le fait d'utiliser un micro-monde comme environnement d'apprentissage est essentiel pour nous. Cet environnement supporte les apprentissages de l'étudiant en permettant la mise en application des concepts présentés. L'interface doit être la plus transparente possible et facile à maîtriser pour ne pas perturber l'apprentissage.

MODÈLE THÉORIQUE

Cette section décrit un modèle théorique pour un système tutoriel utilisant un micro-monde comme interface étudiant-système. Ce modèle incorpore les concepts de curriculum, de modélisation de l'étudiant, de syllabus de cours et d'enseignement. En réaction aux actions de l'étudiant dans le micro-monde, le système oriente sa stratégie pour atteindre les objectifs fixés par le syllabus choisi pour un cours donné.

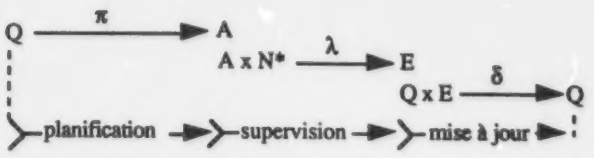
Un **curriculum** est un 5-tuplet $C = (Q, E, \delta, S, \phi)$ où Q, E, S sont des ensembles finis non vides et δ: Q x E -> Q, ϕ: S x Q -> {vrai, faux} sont des fonctions. Q est l'ensemble des *états* du curriculum, E est l'*alphabet d'évaluation*, S est l'ensemble des *syllabi*, δ est la fonction de *mise à jour* et ϕ est la fonction de *fin du cours*. Un état $q \in Q$ décrit le contexte actuel du curriculum du point de vue de la matière à enseigner et de ce que l'étudiant en connaît. Cela inclut entre autres les savoir-faire acquis par l'étudiant tels qu'évalués par le système. Une évaluation $e \in E$ est un symbole utilisé pour représenter le résultat d'une interaction avec l'étudiant. Dans la pratique cet alphabet peut être très étendu. La fonction de mise à jour δ permet le passage d'un état q à un autre état q' en fonction d'une évaluation particulière $e \in E$. C'est pourquoi la fonction δ agit comme une fonction de mise à jour pour le curriculum. Un syllabus $s \in S$ fixe le cours qui sera enseigné par le STI. Le choix d'un syllabus particulier $s \in S$ détermine, par exemple, si le cours est de niveau élémentaire, intermédiaire ou avancé. Ce choix détermine également si le cours s'adresse plus particulièrement à monsieur tout-le-monde, à un ingénieur ou à un administrateur. La fonction fin du cours ϕ vérifie si un état $q \in Q$ rencontre l'ensemble des objectifs du syllabus $s \in S$.

Une **description de cours** est un couple (C, s) où $C = (Q, E, \delta, S, \phi)$ est un curriculum et $s \in S$ un syllabus. Soit une **description de cours individualisée** (C, s, q_0), où $C = (Q, E, \delta, S, \phi)$ est un curriculum, $s \in S$ un syllabus, $q_0 \in Q$ est l'état initial du curriculum pour un étudiant donné. Il est possible d'y associer un automate fini $A_C(s) = (Q, E, \delta, q_0, F)$, où l'ensemble des états finaux est donné par $F = \{q \in Q \mid \phi(s, q) = \text{vrai}\}$.

Un **système tutoriel** est un 5-tuplet $T = (C, A, N, \pi, \lambda)$, où $C = (Q, E, \delta, S, \phi)$ est un curriculum, A et N sont des ensembles finis non vides, π: Q -> A est une fonction et λ: A x N* -> E (où N* est le monoïde libre sur N) est une fonction partielle (c'est-à-dire que le domaine de λ n'est pas tout A x N*). A est l'*ensemble des activités*, N est l'*alphabet des actions* (les symboles de N représentent les noms des actions élémentaires réalisables dans le micro-monde), π est la *fonction de planification* et λ est la *fonction de supervision*.

Associé au système tutoriel T est une fonction partielle Ω: Q x N* -> Q appelée *fonction d'enseignement*. La fonction Ω est définie par

$$\Omega(q, w) = \delta(q, \lambda(\pi(q), w)).$$



La fonction Ω est réalisée en trois phases (figure 1): planification, supervision et mise à jour. La première phase utilise l'état courant q pour déterminer une activité $\pi(q)$. Lors de la supervision de cette activité, l'étudiant exécute une séquence d'actions dans le micro-monde. Cette séquence d'actions constitue un mot $w \in N^*$. Ces actions sont analysées et une évaluation $\lambda(\pi(q), w)$ est produite. Enfin, utilisant le résultat de cette évaluation la troisième phase met à jour le curriculum.

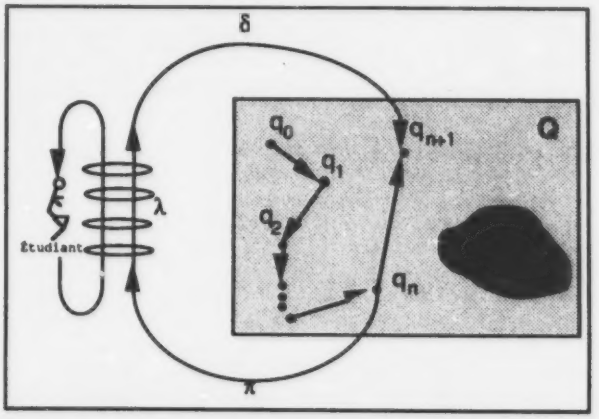


Figure 1: Fonction d'enseignement et Curriculum

Soit une description de cours initialisée pour un étudiant donné (C, s, q_0). Le **cycle d'enseignement** consiste à obtenir une suite $(q_0,w_0), (q_1,w_1), (q_2,w_2), (q_3,w_3), \ldots, (q_{k-1},w_{k-1}), q_k$, où pour chaque q_i un w_i est obtenu de l'interaction avec l'étudiant en utilisant le micro-monde et où $q_{i+1} = \Omega(q_i, w_i)$ (figure 1). L'objectif étant de faire en sorte que q_k soit éventuellement dans F. L'aspect *intelligent* du système tutoriel se retrouve dans la réalisation des fonctions de planification et de supervision.

PROTOTYPE 'WORD TUTOR'

Nous sommes à réaliser un prototype de système tutoriel intelligent selon le modèle présenté. Ce prototype, **Word Tutor** [3,5], comporte les composants **Curriculum**, **Planificateur**, **Superviseur** et **Interface**.

Chaque composant possède une expertise propre qui lui permet de jouer un rôle spécifique à l'intérieur du système tutoriel (planification, supervision, mise à jour…)

et l'intégration des composants constitue un système tutoriel intelligent capable d'enseigner, d'une manière adaptative, un cours complet.

Le Curriculum

Celui-ci contient une représentation de l'expertise concernant l'organisation globale de la matière, une représentation des objectifs d'apprentissage pour un cours particulier (syllabus) et une représentation détaillée des savoir-faire que possède l'étudiant par rapport à la matière enseignée (modèle de l'étudiant). Pour une même matière, le curriculum peut supporter plusieurs syllabi. Il propose, au planificateur, des unités d'enseignement en se basant sur le modèle de l'étudiant et le syllabus choisi pour le cours. Il possède également une fonction qui lui permet de détecter la fin du cours. Enfin, il met à jour le modèle de l'étudiant selon l'évaluation reçue du Superviseur. La structure interne du Curriculum est décrite plus en détails dans [5].

Le Planificateur

Le Planificateur ne connaît pas la matière enseignée. Celle-ci est représentée dans le curriculum (structure) et dans le module expert du domaine. Par contre, il utilise l'expertise du Curriculum et sa propre expertise pédagogique pour établir l'ordonnancement des différentes activités du cours. Cette expertise de planification se veut la plus indépendante possible de la matière enseignée.

Le Superviseur

Le Superviseur génère le contenu des tâches associées aux activités identifiées par le planificateur et coordonne leur présentation auprès de l'étudiant. Pour chaque exercice proposé à l'étudiant ou pour chaque exemple présenté par le Superviseur, les solutions aux questions posées sont fournies par le module expert du domaine, alors que l'évaluation des actions de l'étudiant lors de la réalisation d'un exercice dans le micro-monde est réalisée par le module de diagnostic. De plus, si une intervention est jugée nécessaire par le Superviseur, celui-ci en détermine la nature et son contenu, tandis que l'interface en assure la présentation.

L'Interface

Le micro-monde constitue la principale composante de l'interface de communication entre le Superviseur et l'étudiant. Le micro-monde permet à l'étudiant de pratiquer des savoir-faire dans un contexte qui simule la réalité. Une autre partie de l'interface fournit la possibilité d'un dialogue en langage naturel restreint et met à la disposition de l'étudiant certains *utilitaires*.

Cycle d'enseignement

La fonction d'enseignement Ω considère deux aspects du modèle de l'étudiant: l'aspect *stratégique* et l'aspect *psychologique*. Le premier aspect concerne les savoir-faire que possède l'étudiant et l'utilisation de ceux-ci dans le micro-monde, tandis que le second aspect concerne des caractéristiques personnelles de l'étudiant (formes d'apprentissages préférées, rapidité d'apprentissage, attitude à l'égard du système et de la matière, capacité de concentration...).

La fonction d'enseignement Ω utilise principalement trois composants: le Curriculum, le Planificateur et le Superviseur. La fonction de planification π est réalisée par le Planificateur, la fonction de supervision λ par le Superviseur et la fonction de mise à jour δ par le Curriculum.

La fonction de planification π doit fournir une activité $\pi(q) = (t_1, t_2, t_3, _, t_k)$ constituée d'une suite de tâches à exécuter par l'étudiant dans le micro-monde. Pour réaliser cette fonction, le Planificateur sélectionne un sujet à enseigner, puis prépare la suite des tâches (vidéo, exemples, exercices) constituant la prochaine activité. Les présentations vidéo et les exemples guidés servent d'outils didactiques dans le processus tutoriel. Enfin, le Planificateur envoie au Superviseur cette activité pour qu'elle soit effectivement réalisée.

Lors de la supervision de l'étudiant, le module expert du domaine, le module de diagnostic et l'interface sont utilisés. Dans le cas d'un exemple, le Superviseur génère une question correspondant à un problème type et résout dynamiquement celui-ci pour l'étudiant (c'est-à-dire présente la solution générée par le module expert du domaine). Dans le cas d'un exercice, le Superviseur génère pour l'étudiant une question et observe celui-ci lorsqu'il solutionne l'exercice dans le micro-monde. En se basant sur le résultat de cette observation, le Superviseur décide d'une intervention si la situation le requiert. Si une intervention est requise, le Superviseur utilise l'interface pour la générer dynamiquement. À la fin d'un exercice, le Superviseur utilise le module de diagnostic pour générer une évaluation partielle des actions faites par l'étudiant dans le micro-monde (celles-ci sont des symboles de N dans le modèle théorique). C'est seulement à la fin de l'activité planifiée que le Superviseur utilise toutes les évaluations partielles pour générer une évaluation globale qui est transmise à la fois au Curriculum et au Planificateur (ici, nous dévions quelque peu du modèle théorique tel que présenté).

Le Curriculum utilise cette évaluation pour réaliser la mise-à-jour (fonction δ), tandis que le Planificateur réévalue ses plans en fonction de l'évaluation reçue.

États du système tutoriel

Nous appelons *état du système tutoriel* un objectif local que poursuit le système tutoriel à un instant donné. Les états possibles lors d'un cycle d'enseignement sont: déterminer un plan, déterminer un ensemble de sujets à enseigner, sélectionner un contexte, déterminer une activité, évaluer un plan, ordonner les tâches d'une activité, déclencher une séquence vidéo, générer une question, générer une solution, présenter une solution, observer l'étudiant, décider d'une intervention, réaliser une intervention, réaliser une intervention partielle, réaliser une évaluation globale, réaliser la mise à jour du curriculum.

Les fonctions de planification et de supervision sont réalisées par des **systèmes à base de règles** tandis que la fonction de mise à jour correspond à la fonction de transition d'un automate fini.

CONCLUSION

Nous avons présentés dans cet article quelques aspects du modèle que nous utilisons pour réaliser le prototype **Word Tutor**. Une des préoccupation majeure du groupe HÉRON est de séparer, en autant que possible, les différentes expertises requises pour obtenir un système tutoriel intelligent. À moyen terme nous nous proposons d'utiliser la même architecture (c'est-à-dire la même structure pour le Curriculum, le même Planificateur, essentiellement le même Superviseur (sauf pour le module expert de la matière et le module de diagnostic)) pour réaliser un autre système tutoriel utilisant un nou-veau micro-monde. Le but à long terme étant de réaliser une coquille de système tutoriel intelligent.

Le prototype est réalisé sur un poste de travail SUN à l'aide de l'environnement de développement de systèmes experts ART ("Automated Reasoning Tool"). Le groupe HÉRON bénéficie du soutien des subventions CRSNG thématique (G-1979) et FCAR équipes et séminaires (88-EQ-3627).

RÉFÉRENCES

1. Anderson,J.R. (1988). The Expert Module. Dans Polson,M. & Richardson,J.J.,(Eds.). *Foundations of Intelligent Tutoring Systems*. Hillsdale, N.J: Lawrence Erlbaum Associates.

2. Gagné,R.M. & Briggs,L.J. (1979). *Principles of Instructional Design*. New-York: Holt, Rinehart & Winston.

3. Gauthier,G., Hopper,C. & Imbeau,G.(1987). Le design d'un prototype de système d'enseignement intelligemment assisté par ordinateur. *Les actes du CIPTE-87*.

4. Frasson,C. (1988). Intelligent Tutoring Systems: the HÉRON Project. *Canadian Artificial Intelligence*, pp. 27-29.

5. Imbeau,G.,Frasson,C. & Gauthier,G.(1988). A Multi-Expert System for Large Scale Intelligent Tutoring. Dans *Les actes de la conférence ITS-88*, Université de Montréal, Montréal, pp. 272-278.

6. Papert,S. (1987). Microworlds: Transforming Education. Dans Lawler,R.W. & Yazdani,M.,(Eds.). *Artificial Intelligence and Education, Volume one: Learning Environments & Tutoring Systems*, Norwood, N.J.: Ablex.

7. Stevens, A.L., Collins, A. & Goldin, S. (1979). Misconceptions In Student's Understanding, *International Journal on Man-Machine Studies*, 11, pp. 145-156.

8. Stevens, A.L., Roberts, B. & Stead, L. (1983). The Use of a Sophisticated Interface in Computer-Assisted Instruction. *IEEE Computer Graphics and Applications*, 3, pp. 25-31.

9. Suppes, P. (1981). University-Level Computer-Assisted Instruction at Stanford: 1968-1980, Stanford, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University.

ADAPTATION DYNAMIQUE DE L'ENSEIGNEMENT DANS LES SYSTÈMES D'EIAO: LE CAS DU SYSTÈME SEISDI

A. Benslimane, C.F. Ducateau
Dép. informatique
Institut Universitaire de Technologie de Clermont-Ferrand
Aubière, France

Le but de notre article est d'exposer, à travers le système SEISDI, une méthodologie de prise en compte de l'adaptation dynamique de l'enseignement dans un système d'EIAO. Ce dernier ayant comme méthode d'enseignement la présentation des connaissances d'une matière avec résolution d'exercices par l'apprenant. Les différentes composantes d'un tel système et les formalismes de représentation utilisés seront explicités. SEISDI est un système d'EIAO permettant d'adapter dynamiquement l'enseignement à l'apprenant. Cette adaptation est réalisée par un ensemble de systèmes experts. Ces derniers s'appuient sur une Base de plans d'enseignement, un modèle de la matière, un modèle de l'apprenant et un ensemble de Bases de connaissances. Ces systèmes experts ont pour but: la génération de la Base de plans d'enseignement, le choix et l'exécution d'un plan, la sélection d'exercices et l'orientation pédagogique. Un plan d'enseignement contient un choix des connaissances à présenter à l'apprenant avec un niveau de détail, une forme d'enseignement et les moments de présentation d'exercices. En début de session, un plan d'enseignement est choisi à partir de certaines caractéristiques de l'apprenant. Lorsque le moment de contrôle d'acquisition des connaissances intervient, le module de sélection d'exercices choisit un exercice grâce à sa Base de connaissances. Une fois l'exercice résolu par l'apprenant, les paramètres décrivant son comportement sont saisis et communiqués au module d'orientation pédagogique. Ce dernier les analyse et détermine les actions à entreprendre pour la suite de l'apprentissage (changement de plan d'enseignement, renforcement du plan, etc. ..). La représentation de la matière que nous exposerons est basée sur un modèle spécifique que nous avons défini. Ce modèle met en évidence les différents objets d'enseignement de la matière ainsi que les relations logiques et pédagogiques entre ces objets. La représentation du modèle par un réseau multi-couches permettra, par utilisation d'une Base de connaissances, de générer automatiquement la Base de plans d'enseignement.

DYNAMIC ADAPTATION OF EDUCATION IN ICAL SYSTEMS WITH SPECIFIC REFERENCE TO THE SEISDI SYSTEM

The purpose of our article is to describe, using the SEISDI system, a method for considering the dynamic adaptation of education in an ICAL system, where the teaching approach of the latter is the presentation of knowledge in a field through exercises solved by the learner. The various components of such a system and the representation formalities used will be explained. SEISDI is an ICAL system allowing the dynamic adaptation of the teaching approach to the learner. This adaptation is achieved by a set of expert systems that rely on a base of teaching plans, a model of the subject matter, a learner profile and a set of knowledge bases. The objective of these expert systems is to generate the base of teaching plans, the choice and execution of a plan, the selection of exercises and the educational orientation. A teaching plan contains a choice of knowledge to be presented to the learner, with the degree of detail, a teaching form and the times for presenting exercises. At the start of a session, a teaching plan is selected on the basis of certain learner characteristics. When the time comes for monitoring knowledge acquisition, the exercise selection module chooses an exercise through its knowledge base. Once the exercise has been completed by the learner, the parameters describing his or her performance are recorded and conveyed to the teaching orientation module. This module analyses them and determines the measures to be taken to continue the learning process (change of teaching plan, reinforcement of the plan and so on). The subject matter representation we will describe is based on a specific model we have defined. This model shows the various instructional objects of the subject matter and the logical and educational relationships among these objects. The representation of the model by a multilayer network will, through the use of a knowledge base, allow the automatic generation of the base of teaching plans.

"ADAPTATION DYNAMIQUE DE L'ENSEIGNEMENT DANS LES SYSTEMES D'EIAO : LE CAS DU SYSTEME SEISDI"

Abdellah BENSLIMANE C.François DUCATEAU

Laboratoire d'Informatique
Institut Universitaire de Technologie
Département Informatique
B.P 86 63172 AUBIERE Cedex (FRANCE)

Résumé: Cet article a pour but de dégager quelques principes pour la prise en compte de l'adaptation dynamique de l'apprentissage dans les systèmes d'EIAO. On s'intéresse plus précisément aux systèmes d'acquisition de contenus d'une matière. L'idée de base est l'utilisation de la planification en s'appuyant sur le modèle de la matière et celui de l'apprenant.

INTRODUCTION

L'un des objectifs principaux des systèmes d'EIAO est l'adaptation dynamique de l'apprentissage à l'apprenant. Ceci veut dire que le système doit prévoir à tout moment l'activité qui soit la plus bénéfique à l'apprenant. De plus, cette adaptation doit se faire au niveau du contenu et de la forme (ONL 87).

Le but de cet article est, à travers le système SEISDI, de dégager quelques principes pour la prise en compte de l'adaptation dynamique de l'apprentissage à l'apprenant. SEISDI (BEN 88a) est un système d'EIAO orienté acquisition de contenus de la matière des "structures de données informatiques". Le principe général de SEISDI est d'élaborer des plans d'enseignement à partir du modèle de la matière et des variantes apprenant. Un plan sera évoqué dans une session d'apprentissage et un contrôle de sa pertinence sera effectuée à chaque fois que l'apprenant résoud des exercices. Ce contrôle débouche sur le renforcement du plan, l'abandon de la session, la remise en cause du plan au profit d'un autre plan,etc...

1.PLANIFICATION ET PLAN D'ENSEIGNEMENT

Face à la variabilité des situations d'apprentissage, la planification apporte une solution à la prise en compte de l'adaptation dynamique de l'apprentissage dans les systèmes d'EIAO. Elle a pour rôle d'accomoder l'apprentissage en se basant sur des prédictions sur le comportement de l'apprenant.

Dans Le système SEISDI,la planification est utilisée à deux niveaux. Le premier concerne l'élaboration de plans d'enseignement à partir du modèle de la matière, et des variantes apprenant. Le deuxième niveau concerne le contrôle de la pertinence d'un plan au comportement apprenant. Un plan d'enseignement est défini comme une représentation schématique (HOC 87) permettant de déterminer:

- Les contenus de la matière à faire acquérir à l'apprenant.

- Les scénarios d'enseignement de ces contenus, ainsi que l'ordre de leur présentation. Un scénario définit un niveau de détail du contenu ainsi qu'une forme d'enseignement qui lui est associé.

- Les moments opportuns d'évaluation de l'acquisition des connaissances par l'apprenant.

Remarque: La tâche de sélection d'exercices appropriés au contexte d'apprentissage est éffectuée par un module spécifique intervenant en cours de déroulement de la session. Ceci est fait dans le but de réduire la compléxité d'élaboration de plan d'enseignement.

II. ARCHITECTURE DE SEISDI

Pour répondre aux éxigences de l'adaptation dynamique de l'apprentissage, le système SEISDI assure Trois principales fonctions, chacune d'elle constitue un composant de SEISDI (figure 1):

- L'élaboration de plans d'enseignement réalisée par le module GENPLAN

- L'évocation et execution de plan assurée par le module EXEPLAN

- Le contrôle de la pertinence d'un plan au comportement d'un apprenant. Il est éffectué par le module CONPLAN

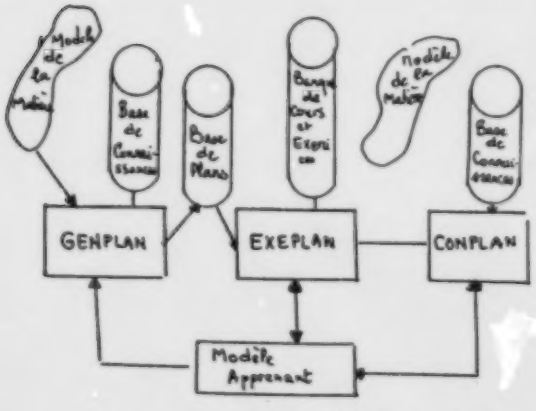


Figure 1: Architecture générale de SEISDI

III. DEROULEMENT DE L'APPRENTISSAGE

Un plan d'enseignement est choisi en début de session pour chaque apprenant par le module EXEPLAN. Ce plan est choisi en fonction du niveau de l'apprenant dans le domaine et de sa capacité d'apprentissage. Le plan est déroulé par EXEPLAN. Lorsque le moment d'évaluation de l'acquisition des connaissances par l'apprenant intervient, EXEPLAN par un sous-module spécifique séléctionne un exercice approprié au contexte d'apprentissage. L'apprenant essaie de le résoudre, et les paramétres décrivant son comportement sont saisis et stockés dans le modèle apprenant. A ce moment, le module COMPLAN intervient et juge l'adéquation du plan avec le comportement de l'apprenant et apporte les corrections si celà est nécéssaire.

IV. LE MODELE DE LA MATIERE DANS SEISDI

Il contient les connaissances de la matière organisées en objets d'enseignement, et les relations entre ces objets (BEN 88b). Les objets d'enseignement sont définis par des attributs (complexité d'acquisition de l'objet, fonction de l'objet dans le processus d'apprentissage, pré-condition de déclenchement, etc...). Les objets d'enseignement peuvent être de type conceptuel ou concret. Un objet conceptuel est un objet décomposable en d'autres objets. Un objet concret est un objet auquel correspond un contenu précis d'enseignement. A un objet concret corresponds plusieurs granules d'enseignement. Ces derniers décrivent le contenu d'un objet concret avec des scénarios différents.
Les relations entre objets d'enseignement sont de type fonctionnel (relation de manipulation, d'opposition) ou de type pédagogique (relation de proximité pédagogique, de rappel, de synthèse, d'inclusion, etc...). La représentation du modèle est éffectuée à travers un réseau multi-couches (figure 2). Chaque couche représente un degré de raffinement de la matière. L'implémentation est réalisée par utilisation de "frames"

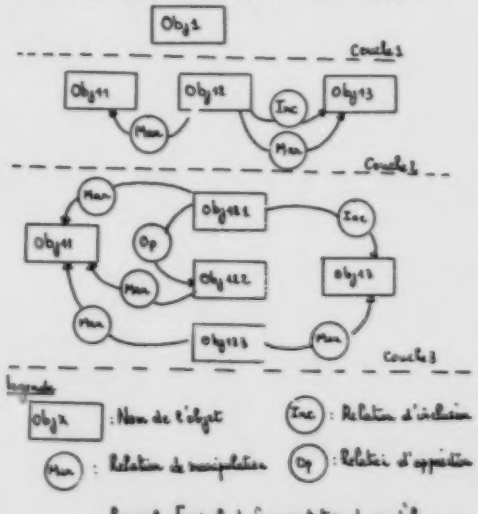


Figure 2: Exemple de représentation du modèle

V. LE MODELE APPRENANT

Il est utilisé dans SEISDI pour le choix d'un plan d'enseignement, et pour le contrôle de la pertinence d'un plan au comportement de l'apprenant. Les informations que l'on distingue sont:

1) Les caractéristiques apprenant: Ce sont le niveau de l'apprenant dans le domaine et sa capacité d'apprentissage

2) Le contexte global d'apprentissage: Il comprend:

- la liste des objets d'enseignement acquis par la pratiqsue du syatème, avec pour chaque objet le degré d'acquisition atteint

- La moyenne des notes obtenues pour chaque type d'exercices. Un type d'exercices détermine l'objectif à atteindre dans la processus d'apprentissage. Le type 1 correspond à des questions de cours. Le type 2 à des applications directes des connaissances véhiculées par des objets d'enseignement. Les exercices de type 3 sont des exercices qui requiérent une habileté de la part de l'apprenant.

3) Contexte local d'apprentissage: Il comprend:

- la description de l'exercice venant d'être résolu par l'apprenant tel l'ensemble des objets d'enseignement véhiculés par l'exercice, sa complexité par rapport à l'apprenant, etc...

- Les informations décrivant le comportement de l'apprenant tel la note obtenue, le temps de réponse éffectué, le type d'erreur, etc...

- Le point de reprise. Il concerne la reprise d'une session d'apprentissage interrompue par l'apprenant. Les informations de reprise concerne le module à relancer et les informations nécéssaires à sa relance.

VI. LES FONCTIONNALITES DE SEISDI

VI.1 ELABORATION DE PLAN D'ENSEIGNEMENT

Elle est éffectuée par le système expert GENPLAN. Il utilise le modèle de la matière, une Base de Connaissances d'élaboration de plans et des caractéristiques apprenant. Les plans générés sont stockés dans une Base de plans.

La Base de Connaissances d'élaboration de plan contient les heuristiques d'élaboration de plan. Ces heuristiques partent de la couche la moins raffinée du modèle et passent successivement aux autre couches. Les heuristiques sont représentées sous forme de régles de production. Ces dernières sont regroupées en quatres catégories. La première catégorie comprend des régles permettant le choix des objets d'enseignement à faire acquérir à l'apprenant. Elles opérant à partir des caractéristiques des objets d'enseignement. Les régles de deuxiéme catégorie ont pour but d'ordonner les opbjets d'enseignement choisis. Elles utilisent pour celà les relations entre objets. Les régles de troisiéme catégorie permettent de choisir les moments d'évaluation de l'acquisition des connaissances par l'apprenant. Ceci est fait à partir de la complexité de l'objet d'enseignement, de la capacité d'apprentissage de l'apprenant, et la fonction assurée par l'objet d'enseignement dans le processus d'apprentissage (rappel, synthése, description de faits,etc...). Les régles de la quatriéme catégorie ont pour objectif de déterminer pour chaque objet d'enseignement le scénario le plus adéquat à l'apprenant. Elles utilisent pour celà de la complexité de l'objet d'enseignement et de la capacité d'apprentissage de l'apprenant.

Les régles de la catégorie 1 et 2 sont évoquées pour chaque couche du modèle de la matiére. Les régles de la catégorie 3 et 4 sont utilisées uniquement pour la derniére couche du modèle de la matiére.

VI.2 L'EVOCATION DE PLAN D'ENSEIGNEMENT

Cette tâche comprend le choix d'un plan d'enseignement en début de session pour un apprenant, et le choix d'exercice approprié lorsque dans le plan choisi intervient le moment d'évaluation de l'apprenant. C'est le système expert EXEPLAN qui assure l'évocation de plan. Il opère à partir d'une Base de Connaissances, d'une Base de plans et du modèle apprenant.

La Base de Connaissances contient un ensemble de règles. Ces règles permettent de choisir un plan d'enseignement à partir du modèle apprenant et de selectionner les exercices à soumettre à l'apprenant. Ces dernières déterminent d'abord le type de l'exercice à soumettre et par la suite l'exercice éffectivement à soumettre. Le type d'exercice est déterminé à partir des caractéristiques apprenant et de l'objet d'enseignement constituant le but de l'évaluation de l'acquisition des connaissances par l'apprenant. L'exercice éffectivement a présenter est déterminé en utilisant la liste des exercices déjà éffectués par l'apprenant.

VI.3 LE CONTROLE DE PLAN D'ENSEIGNEMENT

Il est réalisé par le système expert COMPLAN. Il est activé à chaque résolution d'exercice par l'apprenant. Son objectif est d'analyser le comportement de l'apprenant lors de la résolution d'exercice, et de juger de la pertinence du plan au comportement apprenant. Au cas ou le plan s'avére mal adapté, des corrections seront apportées. Celles-ci sont regroupées en types d'actions à engager. Les types que l'on distingue sont le renforcement de plan (apport d'une explication, redéroulement du plan à partir d'un certain point, presenter un autre exercice, changement de scénario, etc...), la remise en cause du plan au profit d'un autre ou l'abandon de la session. Chacun de ces types d'actions est déterminé par utilisation d'une Base de Connaissances, du modèle de la matière et celui de l'apprenant. Ainsi, par exemple, si l'apprenant n'a pas acquis un objet d'enseignement X qui est en relation de manipulation avec un objet Y, deux possibilités peuvent être déduites par COMPLAN. La première consisterait à faire un rappel de l'objet Y si le système expert juge que l'apprenant a besoin juste d'un rappel. La deuxième possibilité consisterait à remettre en cause le niveau de l'apprenant dans le domaine et par là à la remise en cause du plan en cours.

CONCLUSION

A travers cet article, nous avons tenté de dégager quelques principes pour la prise en compte de l'adaptation dynamique de l'apprenatissage à l'apprenant. Le modèle de la matière qui a été défini, a une portée assez étendue dans la mesure où il peut s'appliquer à des matières hiérarchisées. Le principe de la planification apporte une réponse à la variabilité des situations d'apprentissage. Au niveau du contrôle du plan, il nous semble qu'il faut engager des réflexions sur l'analyse sémantique des erreurs de l'apprenant.
Le système SEISDI est un système pouvant supporter l'apprentissage de plusieurs matières de natures différentes pour peu que les bases de connaissances utilisées soient adaptées.

REFERENCES

(BEN 88a): A.BENSLIMANE C.F DUCATEAU
"Un système d'EIAO basé sur la coopération de plusieurs systèmes experts: le système SEISDI"
1°Congrés Européen Intelligence Artificielle et formation , LILLE 88, France

(BEN 88b): A.BENSLIMANE
"Analyse et modélisation d'une matière"
Rapport Interne de recherche, IUT Aubière, 1988
(NOC 87): J.M NOC
"Planification et Psychologie cognitive"
Presses Universitaires de Grenoble 1987

(OHL 87):S.OHLSON
"Some principles of Intelligent Tutoring System"
Artificial Intelligent and Education Volume One Edited by Yazdani

(SLE 82): D.SLEEMAN J.S BROWN
"Intelligent Tutoring Systems"
Academic Press,INC, Edited by D.SLEEMAN

THE USE OF MICROCOMPUTER BASED KNOWLEDGE REPRESENTATION AS A TOOL FOR LEARNING, TEACHING AND CURRICULUM DESIGN IN PRESERVICE ELEMENTARY SCIENCE EDUCATION CLASSES

M.J. Brody, M.A. McGarry, B. Patterson
College of Education
Unversity of Maine
Orono, ME, USA

This paper describes the use of SemNet* software to help elementary preservice education majors in the College of Education at the University of Maine to learn science concepts, prepare curriculum materials and teach about the environment. SemNet is a microcomputer-based semantic networking tool for the Macintosh computer that permits easy, rapid generation and exploration of multidimensional knowledge structures.

Thirty students were introduced to the use of SemNet in three classes focusing on science education. Instructors in the science classes used SemNet to create multidimensional representations of the science content knowledge. Students in class browsed this network in whatever order their individual needs required, leading to more effective assimilation of new ideas. Students also created their own networks as a study activity leading to accommodation of new ideas and progressive differentiation of existing conceptions. Based on our results, SemNet has proven to be a valuable general purpose thinking and learning tool that is useful at every stage of instruction: analysis, presentation, assimilation and reorganization. It is our belief that computer-based intellectual tools that help students learn new concepts are valuable across disciplines and lead toward higher levels of understanding.

*SemNet is the proprietary trade secret of the SemNet Research Group, a California Limited Partnership - All Rights Reserved.

UTILISATION DE LA REPRÉSENTATION DES CONNAISSANCES SUR MICRO-ORDINATEUR POUR L'ENSEIGNEMENT, L'APPRENTISSAGE ET LA CONSTITUTION DE PROGRAMMES DANS DES CLASSES D'ORIENTATION SUR L'ENSEIGNEMENT DES NOTIONS DE LA SCIENCE

Cette communication a pour objet de décrire comment le logiciel SemNet* peut aider des instituteurs en stage du College of Education de l'université du Maine à apprendre des notions de la science, à constituer du matériel de cours et à enseigner la connaissance de l'environnement. SemNet est un outil de création de réseaux sémantiques tournant sur un micro-ordinateur Macintosh et qui permet une génération et une exploration rapides de structures de connaissances multidimensionnelles.

Trente étudiants ont fait connaissance avec SemNet dans trois classes d'enseignement des sciences. Les instructeurs des classes de science se sont servis de SemNet pour créer des représentations multidimensionnelles de connaissances scientifiques. Les étudiants de la classe ont exploré le réseau dans le sens le plus propice à les conduire efficacement à l'assimilation des nouvelles idées. À titre d'activité d'étude, on a demandé aux étudiants de créer leurs propres réseaux pour qu'ils s'adaptent à cette nouvelle technique et s'entraînent progressivement à différencier les conceptions existantes. D'après nos résultats SemNet a fait la preuve qu'il pouvait être un outil de réflexion et d'apprentissage universel valable, et cela à toutes les étapes de la formation : analyse, présentation, assimilation et réorganisation. Nous sommes convaincus que les outils de micro-ordinateur à réaction intelligente qui aident les étudiants à appréhender de nouvelles notions ont une grande valeur dans toutes les disciplines et peuvent approfondir le niveau de compréhension.

*SemNet est un logiciel secret appartenant à SemNet Research Group, qui est une California Limited Partnership. Tous droits réservés.

The Use of Microcomputer Based Knowledge Representation as a Tool for Learning, Teaching and Curriculum Design in Preservice Elementary Science Education Classes

Michael J. Brody, MaryAnn McGarry, Barbara Patterson

College of Education
University of Maine
Orono, Maine 04469

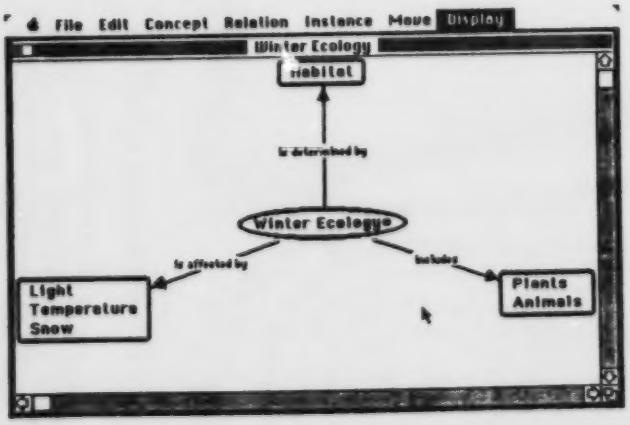
This paper reports on work that is being conducted in the Science Education Program in the College of Education at the University of Maine. The focus is on preservice elementary science education and has a specific pedagogical orientation represented in the educational commonplaces; learning, teaching and curriculum. The three studies are:

1. The Use of Microcomputer Based Knowledge Representation as a **Teaching Strategy** in Preservice Elementary Science Education Classes

2. The Use of Microcomputer Based Knowledge Representation as a **Learning Tool** in Preservice Elementary Science Education Classes

3. The Use of Microcomputer Based Knowledge Representation as a Tool for **Curriculum Design** in Preservice Elementary Science Education Classes.

Each paper focuses on the use of SemNet software (SemNet is the proprietary trade secret of the SemNet Research Group, a California Limited Partnership, All Rights Reserved) to help us better understand student cognition and in particular to improve learning of science content in order to teach science. Other researchers using SemNet are located throughout North America, the Middle East and Australia.



Background

Science education particularly at the elementary level is widely considered to be in a crisis situation. At the national level we must consider our decreasing status in scientific innovation and technological prowess. Science and engineering educational reform efforts are keyed into our national productivity. The National Science Teachers Association has identified elementary science as a key component in a comprehensive plan to improve science education and pre-service science education as a logical place to effect change. Newspapers often publish editorials focusing on the perception on our nation as scientifically illiterate and identify a need for educating the general populace for an economy based on new technologies and information. The big picture is becoming increasingly important. This project focuses on getting perspective teachers to understand science in more meaningful, interconnected ways and to design elementary science course for more effective instruction. We have chosen to address these issues from a cognitive psychological and information processing perspective. It is our belief that we must teach teachers to "learn how to learn" and be able to apply new information in meaningful ways.

In the last half century, there has been increasing interest in the study of memory for educationally useful, meaningful material, and particularly in the relation between "semantic memory structure" or "cognitive structure" and learning (Nagy, 1984). By examining the organizational and structural relationships perceived among concepts, we may learn more about the process of concept acquisition in memory, and consequently may be able to devise more effective instruction (Resnick, 1981).

One of the challenges involved in representing and studying cognitive structure is that it is a dynamic rather than a static entity, subject to change with the acquisition of new knowledge (Sutton, 1980). Another is it is not clear how cognitive structure might be measured or described. A variety of methods have been developed, each based on and resulting in somewhat different assumptions. Many of these methods and their attendant strengths and weaknesses have been summarized in recent books (Nagy, 1984; Holly & Dansereau, 1984; West & Pines, 1985).

Since the constructivist view of learning (Kelly, 1963; Wittrock, 1974) has begun to replace behaviorism as the dominant educational paradigm knowledge representation has been recognized as a

potentially valuable learning strategy. Holly and Dansereau (1984) outline the basic premises of this line of reasoning:

1. The cognitive activities that students engage in when encountering academic or technical learning tasks are of critical importance.
2. These activities can be modified through instruction, training and testing to make them more effective.
3. Teaching strategies or materials should be oriented toward the facilitation of these cognitive activities.
4. Networking and other spatial mapping techniques can be used as vehicles for developing desired cognitive activities.

There are a number of theoretical reasons for believing that computer-based knowledge networks may help students assimilate new information in meaningful ways, help enhance students' ability to recall that information and apply new knowledge in relevant contexts. The basic principles include: reorganization of knowledge, explicitness of relations, depth of information processing, chunking knowledge, conceptual proximity, spatial learning, direction of information processing and ease of use.

Several indications exist that this will be a very fruitful area to investigate and that this is an idea whose time has come. There are at least three other projects based on similar ideas: the Learning Tool, (Kozma, 1986), Knowledge and Information Mapping (McAleese, 1985) and NoteCards (Halsc, Moran & Trigg, 1987).

A primary goal of science education is to educate students from an incomplete understanding of concepts (novice) to a more complete understanding (expert) in an expedient and effective way. The role of pre-service science education is an important component in preparing teachers to tackle this formidable task with their future elementary students. In order to reach this goal with pre-service elementary science education students we must help them:

1) understand particular concepts and methodologies in several subject areas such as physical science, earth science and ecology,
2) help students overcome sometimes significant negative feelings about science and
3) improve their skills in creating a transition from content knowledge to pedagogically correct curriculum materials.

To address these issues we are using SemNet , a microcomputer-based semantic networking tool (currently for the Macintosh computer) that permits easy, rapid generation and exploration of multidimensional knowledge structures in earth science, physical science and ecology classes with elementary pre-service teachers. In particular with SemNet,.....

1. Science instructors create multidimensional externalizations of their knowledge to focus on substance and manipulate ideas without worrying about linear order of expression.
2. Instructors use the nets in class to convey their knowledge to pre-service elementary education students
3. Students browse the instructors knowledge representation on their own in order to meet their own learning needs.
4. Students create nets of their own when they are studying in order to easily reorganize and elaborate upon existing nets as well as use masking features of the nets to review material.
5. Students use their knowledge representations to organize and sequence lesson plans for elementary classroom science instruction.

The Use of Microcomputer Based Knowledge Representation as a Teaching Strategy in Preservice Elementary Science Education Classes

This aspect of our work seeks to determine the effectiveness of SemNet as a teaching tool. Specifically, information is sought to answer the following questions:

1) Does SemNet facilitate a teacher's understanding of relationships among concepts being presented in a science content course, and thus enhance a teacher's expert knowledge?
2) Does SemNet provide a teacher with an instructional vehicle that clearly conveys substantive content and the richness of conceptual connections between concepts to students in a science content course?

Constructivist learning theory, as it has evolved from schema theory and other related theoretical bases in cognitive psychology is replacing behaviorism as the current educational paradigm (Fisher, et al., 1985). Under constructivist theory the task of teachers is to translate expert knowledge contained in a discipline through appropriate curricular vehicles that enable learners and teachers to share meaning. Expert knowledge must be accurately represented with respect to content and structure in the curriculum to enable learners to engage in the process of meaningful learning (Gowin, 1981).

METHODS

The three researchers involved in this study each teach a content course in science to pre-service teachers enrolled in the College of Education at the University of Maine. One teaches physical science, another earth science and the third ecology. Instructors designed expert maps on topics in each of their subjects.

As part of the design process researchers assessed each other's maps for clarity, explicitness, and validity expressed in relationships. Comments were also elicited in writing and verbally about the potential value of SemNet constructed maps as an instructional vehicle. Written and verbal comments obtained from one another and from other instructors were analyzed by the three researchers and assertions about perceived value of SemNet as an instructional tool were derived. From these assertions, value claims were made as to the potential of SemNet for allowing instructors to represent content knowledge and structure accurately and understandably.

To assess interaction of the instructor with maps as a curricular vehicle in the classroom, videotapes were analyzed focusing on the instructor's classroom transactions with students over the content represented in the maps. Assertions about effectiveness of content presentation with SemNet were derived from written self evaluations made by instructors about their classes made immediately after the class and again after viewing the videotaped classroom sessions. Written and verbal evaluations of SemNet's effectiveness as a classroom strategy for sharing expert meaning were made by other members of the research team as well as other College of Education faculty after viewing the videotapes.

DISCUSSION

Preliminary results indicate that use of SemNet is effective in promoting accurate and inclusive representations of science content during the process of instructional design. With a minimal amount of practice, instructors were able to construct both "top down" maps that reflected course content overviews as well as "bottom up" maps that were content specific for a single concept and led into a rich interconnectedness with other specific and subsuming concepts. Maps were judged by other instructors who did not construct them to be a valid means of conveying content and context (illustrated by the propositional connection among concepts) on the science concepts under study.

Analysis of classroom interactions is still in progress, and tapings are planned for classes during the 1988 spring semester. Initial indications are that maps made by the instructor serve as a useful and effective vehicle for conveying content in a meaningful fashion. Initial reluctance or "overwhelmed" feelings appear to be,for the most part, rather quickly overcome; though indications are that relative maturity of the student may be a factor in ready acceptance of such a strategy. Excitement, frustration, and persistence were observed to be some of the feelings expressed by students as they interacted and reacted to content presented as maps. Initial indications are that students tended to find this conceptual approach compelling, thought provoking, challenging, and valuable to their understanding of the content.

The Use of Microcomputer Based Knowledge Representation as a Learning Tool in Preservice Elementary Science Education Classes

The second aspect of our research focuses on the use of SemNet software with preservice elementary science teachers to address the question of meaningful learning. The effectiveness of SemNet as a learning tool can be assessed by focusing on the following cognitive principles: reorganization of knowledge, explicitness of relations, depth of processing, chunking, conceptual proximity, and spatial learning. The existing body of SEMNET literature (Fisher et al., 1987) cites studies confirming the role SemNet plays in addressing these cognitive learning principles.

Instructional strategies which require students to actively reorganize lecture or textual material can improve recall, especially when dealing with more complex subject matter (Melton, 1967; Martin, 1968; Tulvig, 1968,1978; Holly & Dansereau, 1984b; Shimmerlick, 1978.) In using SemNet, students learn to restructure information through the process of constructing individualized, semantic networks. Students must analyze the nature of relationships among concepts before they can successfully connect concepts according to the semantic network mode of representation. In a sense, each students' network represents a physical model of their idiosyncratic, conceptual schema. The semantic network parallels the way the information is stored in their brain.

Methods

Students enrolled in three science education content courses- earth science, physical science, and ecology, in the spring of 1989, were instructed in the use of SemNet software at a Macintosh cluster provided at the University of Maine in Orono. The authors designed their respective six week units on SEMNET. Students will be shown an overview of a six week science unit on SEMNET the first week of classes. Beginning with one major introductory concept, students were asked to develop a weekly semantic network expanding on the concept. In the following classes, instructors provided master maps in which the science concepts introduced were connected. While students become adept at creating their own maps, the master networks served as models to which students compared their own maps for instructional purposes. As new material is taught each week, students selected additional concepts to add to their preliminary semantic network. Adding the new concepts involved revising their original maps. The task of redesigning concept maps each week forces students to concentrate on the structure of knowledge for their particular discipline. Each week's maps represented the students' current level of understanding.

Evaluation procedures focused on scoring students' weekly semantic networks. Numerical points were awarded to each map and separate score sheets for each of the cognitive principles were maintained. Scoring criteria included: number of correct relationships, level of detail, and degree of hierarchy. Numerical scores, recorded over the six week period of the unit, allow monitoring of individual students' progress through time in each particular cognitive area. In addition, correlation among the cognitive principles will be assessed. Discovering to what degree cognitive principles are related helps determine if a student's success or improvement in one cognitive area necessitate success in another?

Discussion

From preliminary pilot studies with concept mapping and computer assisted instructional programs, it is expected that students will progress from novice to expert or will demonstrate an increasing level of

competence in designing semantic networks on the computer. In part, the success of SEMNET is due to students' higher level of motivation as they are given control over their own learning. Once they become adept at using SEMNET, they gain a certain freedom in being able to select the level of difficulty and challenge that matches their ability.

Use of Microcomputer Based Knowledge Representation as a Tool for Curriculum Development in Preservice Elementary Science Education

Course design is an important concept for preservice teachers to understand and a valuable skill which can help lead to their future success in teaching. This paper describes the use of SemNet software to help students restructure scientific knowledge into manageable units leading to the planning and sequencing of appropriate learning outcomes for elementary students.

The concepts of curriculum and instruction are critical for our understanding of the course design process. According to Posner and Rudnitsky (1986) instruction is considered a process- a series of events intended to lead to some learning outcomes. Curriculum on the other hand is not a process but a set of intentions or a set of intended learning outcomes (Johnson, 1967). This approach draws upon the constructionist perspective and supports the constructivist view of learning (Kelly, 1963; Wittrock, 1974). This leads to the analogy of course design as a process much like that of house construction. The curriculum or intended learning outcomes can be considered the blue print or architectural design and instruction the process of building the house. Therefore, course design must first deal with the selection and organization of a set of intended learning outcomes which are directly related to the structure of knowledge. Instructional decisions follow based on the nature of the learner and classroom.. In this study, we focus on the relationship of knowledge representation and the selection of intended learning outcomes. We attempt to answer the question: How does the restructuring of content knowledge affect a preservice education majors selection and sequence of intended learning outcomes for teaching science to elementary students?

Discussion

Preliminary results indicate that students' reorganization of the content knowledge over time was extensive and this led to correspondingly extensive changes in the conceptualization of ILOs. Among the strategies that students employed were 1)chunking science concepts into manageable units which led to fewer and more generalized ILOs, 2) changes in the direction of cognitive processing from more inclusive to specific concepts (and the reverse) in sequential versions of the nets, and 3) the proximity of concepts in the maps led to corresponding sequential plans for the introduction of concepts in classrooms.

Conclusions

The results of the three studies reported in this paper lead us to conclude that the use of SemNet software on the Macintosh computer facilitates the reconstruction of scientific knowledge by teachers and learners in a preservice elementary education program. In addition, it is clear that a better understanding of both content and the process of reconstruction allows pre-service teachers to better conceptualize what science they plan to teach in their elementary classrooms.

References

Fisher, K.M. et al. 1985. Computer Based Knowledge Representation as a Tool for Students and Teachers. AERA paper set presented at Annual Meeting of AERA, Chicago, III.

Gowin, D. Bob. 1981. Educating. Ithaca, New York: Cornell University Press.

Halasc, F., G. Moran, T. P., & Trigg, R. H. (1987) NoteCards in a nutshell. In the Proceedings of the Association for Computing Machinery's Conference on Computer Human Interaction and Graphics Interfaces. Toronto, Canada.

Holly, C.D. & Dansereau, D.F. (1984). Spatial learning strategies. San Francisco: Academic Press.

Johnson, M. (1967). Definitions and models of curriculum theory, Educational theory, 17, 2, 127-139.

Kelly, G. A. (1963). The psychology of personal constructs (Volumes I and II). New York: Norton.

Kozma, R. (1986). personal communication.

Marin, E. (1968). Stimulus meaningfulness and paired associated transfer: An encoding variability hypothesis. Psychological Review, 75, 421-441.

McAleese, R. (1985). Some problems in knowledge representation in an authoring environment: exteriorization,anomalous state meta-cognition and self confrontation. Programmed learning and educational technology, 22, 4, 299-306.

Melton, A.W. (1967). Repetition and retrievel from memory. Science. 158, 532.

Nagy, P. (Ed.). (1984). The representation of cognitive structure. Toronto: Department of Measurement, Evaluation and Computer Applications, The Ontario Institute for Studies in Education.

Novak, J & Gowin, D.B. (1984). Learning how to learn. Cambridge, England: Cambridge University Press.

Novak, J. (1977). A theory of education, Ithaca, NY, Cornell University Press

Posner, G. & Rudnitsky, A. (1986). Course design: A guide to curriculum development for teachers. New York and London, Longman Inc.

Resnick, L. B. (1981). Instructional psychology. Annual review of psychology, 32; 659-704.

Shimmerlick, S.M. (1978). Organization theory and memory for prose: A review of the literature. Review of Educational Research, 48, 103-120.

Sutton, C. R. (1980). The learner's prior knowledge: A critical review of techniques for probing its organization. European journal of science education, 2, 107-120.

Tulvig, E. (1968). Theoretical issues in free recall. In T.T. Dixon & D. L. Horton (Eds.) Verbal behavior and general behavior theory. Englewood Cliffs, NJ: Prentice-Hall.

Tulvig, E. (1978). Relation between encoding specificity and levels of processing. In L. S. Cermak & F. I. M. Craik (Eds.) , Levels of processing and human memory. Englewood Cliffs, NJ: Prentice-Hall.

West, L. H. T. & Pines, A. L. (1985). Cognitive structure and conceptual change. New York: Academic Press.

Wittrock, M. C. (1974). Learning as a generative process. Educational psychologist, 11, 87-95

DIMENSIONS OF INTELLIGENCE IN INTELLIGENT AUTHORING SYSTEM FUNCTIONS

M. Szabo
Dept. of Educational Psychology
Universty of Alberta
Edmonton, AB

M. Allen
Dept. Educational Psychology
University of Minnesota
Minneapolis, MN, USA

Artificial Intelligence is an attempt to program microprocessor systems to perform cognitive operations which require intelligence (thought, reasoning, problem solving and judgment). Progress in AI has been slow except perhaps in the sub-discipline known as Expert Systems, the pursuit of microprocessor-based systems which capture the essence of a human expert in a specific domain (e.g., chess) as a series of content statements bound together by a series of "if.., then..," (production) rules. A system which possesses "intelligence" and is used for courseware authoring is called an Intelligent Authoring System. But what does intelligent mean in this context? How does a more intelligent authoring system behave compared with a less intelligent one? The concern of this paper is to examine from an applied viewpoint indicators of intelligence as represented in several key functions of computer based instruction courseware development. Our starting point is to treat intelligence as a subjective continuum which ranges from less intelligence to more. A subjective transition point between the two is included. Next, 15 categories of courseware intelligence (e.g. Instructional Strategy) and 16 categories of authoring aids intelligence were abstracted from an extensive study of IAS. Each category has been further expanded into a one-dimensional set of subpoints and placed along the intelligence continuum. Examined in this arrangement, the historical, current, and future trends of these categories can be compared. The value in this study is to provide, reflected in the backdrop of current and historical authoring practices, a view of what Intelligent Authoring Systems of the future will look like.

DIMENSIONS DE L'INTELLIGENCE DANS LES SYSTÈMES AUTEURS INTELLIGENTS

L'intelligence artificielle est un essai de programmation des microprocesseurs en vue de faire exécuter à ces derniers des opérations cognitives, c'est-à-dire qui exigent de l'intelligence (réflexion, raisonnement, résolution de problèmes et jugement). Les progrès en IA n'ont pas été très nombreux sauf dans la sous-discipline appelée Systèmes experts, dans laquelle on réalise avec des microprocesseurs des systèmes qui reproduit l'essentiel du comportement d'un expert humain dans un certain domaine (par ex. les échecs) au moyen d'une série d'instructions informatives reliées à une série de règles de décision (genre "if.., then..,). Un système doté d'"intelligence" que l'on utilise pour créer des didacticiels a pour nom Système auteur intelligent. Dans un tel contexte, que signifie intelligent? Comment réagit un système auteur plus intelligent par rapport à un système moins intelligent? L'objectif de cette communication est d'examiner d'un point de vue pratique des indicateurs de l'intelligence que l'on retrouve dans plusieurs fonctions clés du développement de didacticiels. Notre point de départ est de traiter l'intelligence comme un continuum subjectif allant de moins intelligent à plus intelligent; de plus, nous précisons un point de transition subjectif entre deux catégories. Ensuite, nous présentons 15 catégories d'intelligence dans les didacticiels (par ex., stratégie d'enseignement) et 16 catégories d'intelligence d'aide aux auteurs que nous avons dégagées d'une étude exhaustive des SAI. Chaque catégorie est ensuite décomposée en un ensemble unidimensionnel de points qui sont reportés sur le continuum de l'intelligence. En rapportant à cette construction les tendances passées, présentes et à venir de ces catégories, il est possible de les comparer. La valeur d'une telle étude est de dégager, en regard des techniques de création de cours passées et présentes, un aperçu de l'avenir des systèmes auteurs intelligents.

DIMENSIONS OF INTELLIGENCE IN INTELLIGENT AUTHORING SYSTEM FUNCTIONS

Michael Szabo and Michael Allen

University of Alberta and Authorware, Inc.

BACKGROUND

The purpose of this paper is to examine in some depth the juxtaposition of the domains of artificial intelligence, expert systems, and computer based instruction. The quantitative approach chosen examines the use of tools to create (author) computer based instruction materials from the point of view of levels of intelligent functioning.

As a simple example of intelligence, if an authoring tool can anticipate the next move of the author and automatically make that move happen, we might agree that some high level of intelligence is present. A lower level of intelligence might be represented by an authoring system which 'knows' the most likely moves following a given action and makes it simple for the author to execute the desired move. A less intelligent system wouldn't have a clue but would simply list all the options available at that point for easy selection.

Artificial Intelligence is the study of human intelligence through programming machines to emulate human behaviours which appear to require intelligence, such as logic, decision-making, pattern recognition, etc. An *expert system* is the application of AI technology to assume some intelligent decision making function which is of practical value in a narrowly defined range or microworld. *Computer based instruction* (CBI) is the application of computer technology to the process of instruction, teaching, and learning.

The logical link among these three is that they all require relatively high levels of intelligent performance to be successful. Furthermore, they all use computer hardware and software. This gives them at least the appearance of a relationship. The link we have explored and report in this paper is how the application of AI/ES ideas can be made to the process of helping people acquire knowledge, skills, and expertise.

A cautionary note is in order here. Some writers (e.g., Dreyfus and Dreyfus, 1986) have forcefully argued that Expert Systems are not: that is they are merely competent systems. By developing them, we may be confusing competent with expert performance, and unknowingly accepting the former for the latter. Dreyfus and Dreyfus argue that todays' Expert (read Competent) Systems use production rules to make decisions. Experts, they claim, go beyond production rules and use pattern recognition and 'intuition'; two skills which cannot be accommodated by today's computers.

Perhaps their arguments are valid. The point in bringing them out here is to justify our approach of applying intuition to examine intelligent dimensions or features of CBI authoring and courseware delivery. One presumption we make is that the 65 years of CBI experience shared by the authors provides if not expert at least a competent platform from which to begin.

METHODOLOGY

Our first task was to identify important and unique dimensions of authoring aids intelligence to create CBI courseware Generalized intelligence in this area is appearing slowly but there are interesting pockets of intelligence present in some specialized dimensions. We identified sixteen dimensions of authoring aid intelligence: analysis; curriculum design; content, instructional strategy, and student model knowledge acquisition and representation; sequencing; text, speech, graphics, and external device presentation; question generation; input

evaluation; screen management; performance evaluation and courseware evaluation.

In the process we also identified a parallel set of dimensions which we call courseware intelligence aids as they apply more to the interaction between student and courseware (delivery) than to the interaction between author and courseware. The two sets of dimensions parallel one another with a few exceptions. Analysis and curriculum design are specifically authoring functions for which there are no comparable courseware activities or entities. In the area of knowledge acquisition, which was subdivided into content, instructional strategy, and student model knowledge acquisition for the authoring aids review, the study viewed knowledge representation as the corresponding function.

In the next task we recast the notion of intelligence into an intelligence continuum for each dimension. To make the continua most useful for properly characterizing authoring aids, examples of intelligence levels are noted at various points along each continuum, as in Figure 1.0 below.

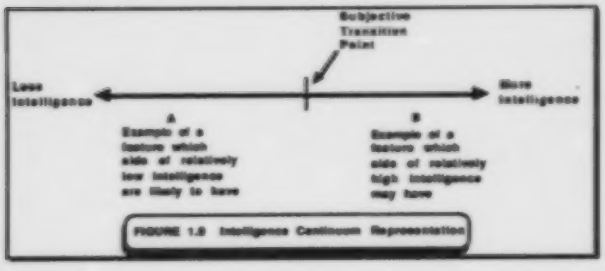


FIGURE 1.0 Intelligence Continuum Representation

In defining these criteria, two sources of intelligence were considered: (1) intelligence in the tool technology itself and 2) intelligence with respect to the process of authoring. 3-D graphics editors, for example, have more advanced tool technology than 2-D graphics editors. Checklist procedures for curriculum definition, on the other hand, have more intelligence with respect to authoring than do the more technologically advanced outliners or "idea processors" which do not recognize subtasks specific to curriculum development. When in conflict, intelligence with respect to authoring activities was given precedence over tool technology intelligence and placed further to the right on the continuum. All items noted are presented simply as examples and reference points. These items by no means comprise exhaustive listings of features currently to be found or predicted for future development. Also, the vertical tick mark on the continuum is at a subjective point where, if an aid has most of the capabilities listed to the left, it might be said to be an "intelligent" aid. The authors do not accept the emphasis on these transition points as denoting the notion of a binary state of intelligence.

SOME EXAMPLES

Below are examples, one each from the authoring and courseware intelligence aids figures which will be presented in the full report. Presented are the Instructional Strategy Definition Aids Intelligence Continuum (Figure 5.0) and the Courseware Graphics Presentation Intelligence Continuum (Figure 24.0).

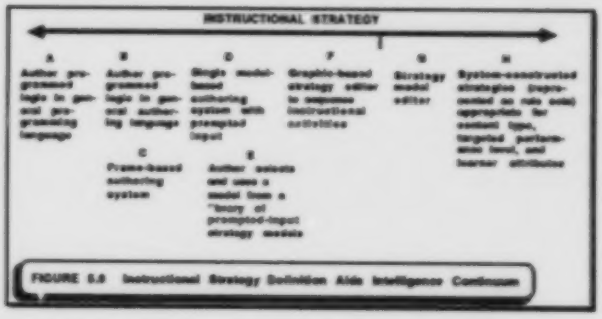


FIGURE 5.0 Instructional Strategy Definition Aids Intelligence Continuum

The definition of instructional strategies is a critical activity not recognized by most authoring aids. The author is expected to have the strategy well conceived and documented so that it can be implemented without many changes. In generalized languages (A) and in most authoring languages (B), the author uses basic conditional branching statements embedded within and distributed throughout the courseware to create functional strategies.

Frame-based systems (C) provide frames specialized as to function. There will be presentation, question, feedback, and menu frames in most systems. Sequencing is performed by branching to another frame. Complex instructional strategies can be very difficult to build and modify because so many branch statements become distributed throughout the courseware. Some systems provide cross-reference tables, circular branching detection, and unresolved (branching to a non-existing frame) checks to help.

Frame-based systems tend to inhibit production of desirable courseware structures.

Frame-based systems can be made easier to use if an instructional strategy is predesigned and the author is prompted in a form-fill-out manner (D). A logical extension is provision of multiple predesigned models (E).

A major advancement is made in providing an editor specifically designed to deal with the instructional strategy. This is more than providing a cross-reference table as in (C), and provides a manner of expressing and communicating about instructional strategies. It seems logical and natural to deal with strategies as a map or other graphic charting means (F). Using this capability not just to build a course but to generate models, perhaps for form-fill-out as in (D), moves into an area of intelligence about courseware designs. Still, the author remains the primary source of instructional strategy design intelligence.

Moving to system-designed, or at least system-selected, strategies (H) represents an appropriate target of advanced intelligent computer based instruction research. Since strategy design is an area of great complexity, creating computerized means of dealing with this issue could have profound advantages.

Under to topic of graphics, nearly all CAI and intelligent CAI systems display graphics in courseware delivery--at least static graphics (Figure 24.0: A). The ability to draw graphics based on state variables (B) is frequently important to simulate processes or display student performance data. While color displays can be very attractive and are sometimes necessary to adequately present some content, the instructional value of color is unclear. It is the opinion of the researchers that color is not an important determinant of courseware intelligence, but were it to be included, each of the example items on the continuum would be duplicated with the monochrome capability to the left and the color capability immediately to the right.

Animations (C) portray much more information than static or even drawn graphics when appropriate to the subject matter. Animations are of many types: moving characters or pictures in a linear, curved, free form, or calculated path; substituting alternative

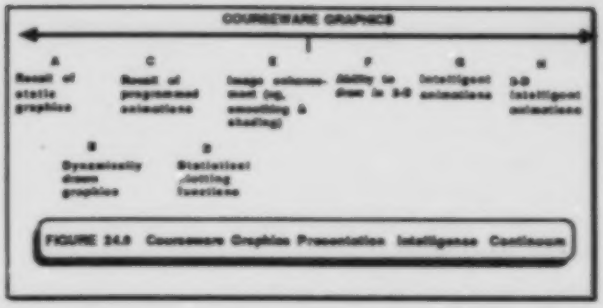


FIGURE 24.0 Courseware Graphics Presentation Intelligence Continuum

characters or pictures during movement; moving one or more points of a line-drawn figure; rotating a figure on a fixed axis; rotating a figure while it is moving along a path; etc. We put three dimensional intelligent animations (H) at the high end of the continuum, but note that simple animations are often the first enhancement once a delivery system is capable of dynamically drawing graphics.

Statistical plotting functions (D) are important to demonstrate mathematical relationships. Again, there are many forms of charting and many mathematical functions that can be supported. At the lowest level, we refer to the ability of the courseware to scale datapoints within a charting area.

Image enhancements (E) such as polynomial smoothing and shading come next. This is often done with color tinting to make shadows and light sources appear very real. Going further, ability to use perspective to create the appearance of three dimensions (F) in dynamically created graphics can enhance the realism of illustrations or more clearly present concepts having spatial relationships. Few authoring systems have such capabilities today, but the attractiveness of this feature and the rapid pace of related technology advancement assure its wide availability in the future.

SUMMARY

This research has examined the current state of the art of application of intelligence to the authoring of CBI courseware. The area of CBI is beginning to grow and gain momentum as education, government, and commercial agencies begin to understand its benefits and as authoring aids become more 'intelligent' and pervasive. We have identified current state of affairs in key dimensions of intelligent

authoring aids and have pointed out valid future directions for profitable and needed growth.

In the limited time available, some examples have been demonstrated of leading edge components, including dynamic (author-driven) creation of instructional strategies, featuring relatively 'intelligent' ways of creating and modifying whole instructional flowcharts as well as smaller components (models). Other features were demonstrated, including transporting courseware for delivery across hardware/software configurations and driving external devices. These should provide evidence of where we are in this field and provide a base from which the those in the field can make further advances.

What of the future of intelligent computer based instruction? We would encourage careful consideration, in proper proportions, of three important considerations. First is the technology of AI/ES itself: the hardware and software which processes inferences rather than numbers. Our analysis has not considered this issue at all for the simple reason that it is here that most AI developers are exclusively focussed. The second area is appropriate models to guide the effort. Our analysis began with the question "What characterizes intelligence in authoring and courseware curriculum development efforts in the first place?" It is akin to carefully casting the problem into a valid form prior to developing solutions. The third consideration is to maintain a healthy respect for current and projected limitations of AI/ES. Intelligence means to ultimately incorporate the intuitive decision making, pattern recognition, and abstraction of relevant information from the mass of date available to us into our performance. We should not settle for systems which are (only) capable of firing a series of production rules regarding quantifiable entities, for the appearance of intelligence therein is deceiving.

REFERENCE

Dreyfus, H. L., & Dreyfus, S. E. (1986). Mind over machine. New York: The Free Press.

A KNOWLEDGE-BASED MULTIPLE-VIEW SIMULATION FOR TRAINING

A.M. Buell, W.D. Godfrey, R. Shingles
Softwords
A Division of Press Porcepic Ltd.
Victoria, BC

A multi-faceted program has been developed for the Canadian Department of Transport to train technicians on how to maintain and calibrate Instrument Landing Systems.

This program presents an interactive graphics-intensive simulation of various views of the ILS system. Both practical and theoretical information about a particular ILS system can be displayed. The internal model of the ILS system is partly mathematical written in the NATAL authoring language and partly expert-system based. As the trainee attempts to perform the various calibration procedures and diagnose difficulties, the inference engine monitors both the trainee's performance and the current state of the ILS system. At any time, the trainee may ask the expert system for advice on how to proceed or get an explanation as to why a particular parameter is out of alignment.

By combining traditional CBT, graphic simulation, knowledge-based and inferencing approaches, a full system has been created which covers a very wide range of needs of the trainee. The approach is applicable to many other situations where skilled technicians are required to improve their understanding and manipulation of complex technical systems.

SIMULATION DE VUES MULTIPLES BASÉE SUR LES CONNAISSANCES POUR L'APPRENTISSAGE

Le Ministère des Transports du Canada a développé un programme polyvalent en vue de former ses techniciens sur la façon de maintenir et d'étalonner les systèmes d'atterrissage aux instruments.

Dans ce programme, on retrouve une simulation infographique intensive de diverses vues du système ILS. On peut faire afficher les informations aussi bien pratiques que théoriques sur un certain système ILS. Le modèle interne du système ILS est en partie un programme mathématique rédigé avec le langage auteur NATAL et en partie un programme basé sur un système expert. Quand un stagiaire désire effectuer les diverses procédures d'étalonnage et diagnostiquer des problèmes, la machine à inférence contrôle autant la performance du stagiaire que l'état actuel du système ILS. Le stagiaire peut toujours demander au système expert de le conseiller sur la façon de procéder ou de lui expliquer pourquoi un certain paramètre est désaligné.

En combinant l'enseignement traditionnel, la simulation infographique, les démarches cognitive et à inférence, on est parvenu à créer un système complet qui peut répondre à une vaste gamme d'exigences du stagiaire. Cette façon de procéder est applicable à de nombreuses autres situations dans lesquelles il faut amener des techniciens expérimentés à comprendre et à dépanner des systèmes techniques complexes.

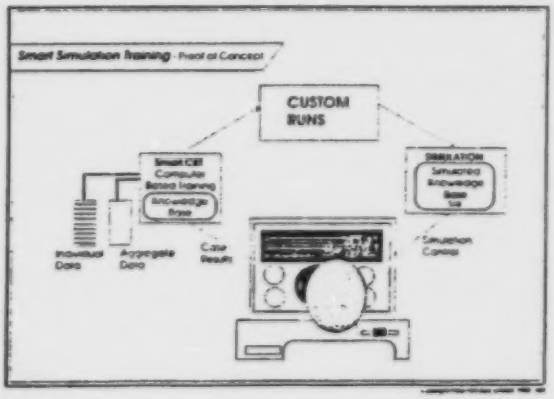
A KNOWLEDGE-BASED MULTIPLE-VIEW SIMULATION FOR TRAINING

A.M. Buell, Dr. W.D. Godfrey, R. Shingles

SOFTWORDS
A Division of Press Porcepic Limited
Victoria, B.C.

INTRODUCTION

Softwords recently completed a multi-faceted computer-based training (CBT) program for the Canadian Department of Transport to train technicians how to calibrate the Instrument Landing System (ILS) equipment. The program combines conventional CBT techniques with graphical simulation driven by knowledge-based representation and inferencing approaches to produce a multiple-view training package. The training problem at the Department of Transport is recurring, deals with complex technology and can only be solved through the specialized training provided by this integrated computer-based training approach.



RATIONALE

Each quarter the ILS equipment must be checked for proper alignment. An in-depth calibration check is conducted annually and takes about twice as long as the quarterly checks. With each alignment check, a specially-equipped calibration aircraft makes survey runs through the ILS envelope. The ground technician makes adjustments to the equipment based on information received from the calibration aircraft.

The training is for the ground technician who makes the adjustments to the ILS equipment. He or she must understand the effects his or her actions have on the ILS envelope each time an adjustment is made. Too often an adjustment takes the system further out of alignment which requires more time and survey runs to finally get the system calibrated. The time the aircraft stays in the air is very expensive.

One faulty adjustment to the equipment could cost DOT an hour's work - about $1500.00. If the plane has to land and the air technician has to correct faulty adjustments, it costs DOT $3000 per hour. If two one-hour delays occurred once a year at each ILS installation it would cost DOT close to $1 million.

The multi-view simulation trainer helps develop the technician's mental model of the system and an understanding of how changes in the equipment settings effect the ILS envelope. The program has the specific goal of training skilled technicians to be effective in the calibration procedures at all ILS-equipped Canadian airports.

THE HARDWARE AND SOFTWARE

On the Sun workstation, the user interface is controlled by icons. All icons on the screen at a given time are accessible to the user and when invoked, operate as an independent process within a unique window. The trainee, once qualified, can review instructional procedures, concepts or definitions while running flight checks or the simulation.

The multiple-view simulation trainer for ILS equipment calibration is developed on the SUN-3/60 color workstation. The graphical simulation is written in Postscript with the interface designed in the object-oriented NeWS windowing system. The Advisor function of the Simulation is represented in KAM/OCCAM (a Softwords-developed C-based knowledge acquisition module and expert system shell) and presented through NATAL. The knowledge bases for the instruction and test were also developed in KAM (Knowledge Acquisition Module, (c) Softwords) and delivered through NATAL.

THE TRAINING

The ILS Equipment Calibration trainer presents a interactive graphics intensive simulation of multiple views of the ILS system. Depending on the trainee's needs and level of expertise, both practical and theoretical information about a particular ILS system is displayed.

The multiple-view training is presented in four main modules:

1) The Help Module
2) The Instruction Module
3) The Simulation Module
4) The Flight Check Module

The training combines conventional CBT in the Help and Instruction modules with the graphical simulation in the Simulation and Flight Check modules. All modules except the Simulation module use knowledge bases to represent the pedagogical or technical information required.

Each module contains a "soft-gate" to the next module. The gate is in the form of a test, the results of which are linked to the trainee registration file. The trainee is able to access the next module once the test for the current gate is complete. From the tests, formative evaluation data is gathered and analysed regarding the effectiveness of the training.

The training interface is exclusively controlled by mouse input once the trainee has signed into the program.

1) The Help Module

The Help Module includes a section on "How the Course Works" and a soft-gate to the next module. "How the Course Works" provides the trainee with a dynamic graphic and text interaction describing the elements of the training program and the mechanics of the multi-windowing interface of the Sun workstation. The module makes use of the Sun NeWS windowing system to show the student dynamic examples of what is being discussed.

In addition to "How the Course Works," there is an embedded help function that the trainee can invoke any time during the training. The center button on the mouse allows easy access to help which is determined by the position of the mouse when clicked. The imbedded help provides information about the function and meaning of the window or item within the window and the relationship between it and other windows or items.

2) The Instruction

The Instruction module includes instruction on all the concepts, definitions, procedures, hardware and formulae related to the training The content of the instruction is stored and organized in a pedagogical knowledge base. The test content is also stored in a pedagogical knowledge base from which questions are automatically generated. The items in this module are accessible from the simulation and flight check modules at any time.

3) The Simulation Module

The core of the training is in the simulation which contains six different representations of the same data. In the field, the technician only has one main indicator from which to make diagnostic conclusions. To provide a well rounded model, the simulation provides the various points of view described below.

3.1) Transmitter

The transmitter window contains the controls that the technician adjusts for calibration. The numerical values generated from the slider settings are used to generate graphical representations used by the trainee to diagnose and correct misalignment.

The transmitter contains five sliders; one each for the identification level, power control, modulation balance, Carrier Sidebands and Sidebands Only phases, and width. Any change in any one of these slider values will result in a change in one or more of the graphical representations described below.

3.2) Power Divider

The power divider window represents the elements of the chosen antenna array as as series of vectors. The trainee has the choice of selecting one of 36 different vector representations. Changes to a vector element can be altered by using the mouse to drag the element to a new position. The effects of these changes are reflected in the other simulation windows.

3.3) Radiation Pattern

This window displays an image of the modulated radiation pattern produced by the system. The radiation pattern is used to locate the position of the calibration aircraft. The horizontal cross-section of the generated pattern is displayed in a combination of blue, red and green representing the carrier, the 90 hertz component and the 150 hertz component, respectively. The pattern reflects changes that occur in the system through transmitter slider adjustments and array element changes.

3.4) Monitor

The monitor window displays the actual monitor panel and its elements. It lets the trainee select from the 19 different monitor modes available in real life. This allows for the representation of alarm limits for misalignment. In real life, the monitor is the only visual aid the technician has to diagnose changes in the system. The combination of the real life monitor and the graphical representations like the radiation pattern provide the trainee with more meaningful information.

3.5) Plane and Truck

The plane window represents the data available to the technician in the calibration aircraft. Similarly, the truck window represents the instrumentation in the far-field truck.

3.6) Carrier Sidebands and Sidebands Only

There are two windows to represent the output of the Carrier Sidebands and Sidebands Only. These two windows contain graphical representations of vectors, bars or sine waves that give the trainee the choice of representation best at the time.

3.7) Advisor

The Advisor combines the inferencing and knowledge representation of the training with the graphical simulation. The Advisor acts as someone who provides the trainee with advice before the ultimate flight check test with the calibration aircraft. When the simulation is first displayed, the trainee must take it out of alignment to see the resultant visual changes. When the trainee selects the Advisor icon, advice is provided about how to recalibrate the system.

The trainee can consult the advisor for a hint, answer or actual calibration. With the Hint option, the advisor will check with the expert system to determine the nature of the misalignment. The Advisor displays a message as to the course of action the trainee should pursue. All adjustments must be made by the trainee.

With the Answer option, the Advisor will check with the expert system as to the nature of the problem as with Hint. The Advisor reports to the trainee the sequence of steps necessary to calibrate the system. The trainee performs the required steps using the mouse.

The Calibrate option of the Advisor will automatically calibrate the system for the trainee. The Advisor consults the expert system to determine the correct sequence of steps and displays them to the trainee as they are executed by the system. The Advisor stacks the errors and presents them in order of importance. The graphic patterns are also updated so the trainee can see the visual effects of the calibration process.

3.8) Test

Once the trainee has practiced and explored the effects of changes in the system, he or she must take the simulation test in order to proceed to the next module. The trainee is presented with a series of misalignments The trainee must calibrate the system given the preset conditions. The Advisor is unavailable to the trainee during the test.

4) The Flight Check Module

A more complex way the program uses the expert system with the simulation is with Flight Checks. A flight check is the ultimate test for calibration and understanding of procedures in which the program simulates the pass that a calibration aircraft would make. The expert system controls the simulated communication from the flight crew overhead to the technician.

In the flight check there are three main actions. The trainee can:

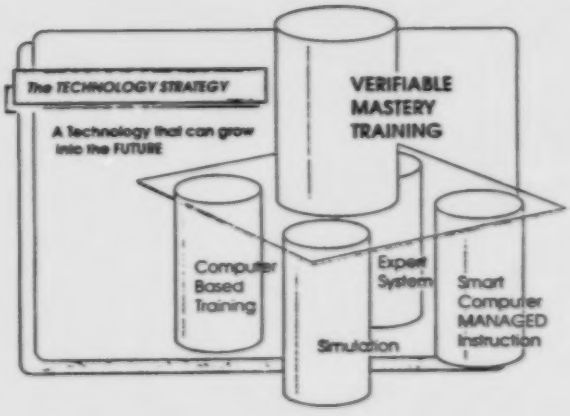
4.1) Select a Scenario
4.2) Select a Test
4.3) Start a Run

The trainee first selects a scenario which is a pre-built set of starting conditions with some misalignment. The trainee then selects the first test using the "Select a Test" option. The tests must be performed in the order presented for correct results. Once the scenario and test are selected, the trainee "Starts a Run." The trainee's task is to make the necessary adjustments to put the system into alignment based on feedback from the aircraft.

Once the run is started, the plane communicates via the expert system to the trainee the adjustments that need to be made to calibrate the system. The trainee is unable to access the Advisor during flight checks.

SUMMARY

According to the research conducted at Softwords, this is the one of the first examples of highly integrated conventional computer-based training and procedural programming techniques with numerical graphical simulation and expert system technology.



As performance data is gathered and analysed for the ILS equipment calibration trainer, the method of integrating multiple programming, presentation and data representation techniques will become more refined. With friendly powerful workstations like the Sun-3/60 becoming more cost effective, integrated computer-based training is a more practical, feasible, effective method of conducting training for complex systems.

UTILISATION DE SYSTÈMES D'EAO DANS DES SYSTÈMES D'EIAO

S. Bengio, C. Frasson, J. Gecsei
Dép. d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, PQ

L'enseignement intelligemment assisté par ordinateur (EIAO) est encore sur des voies de recherches alors qu'une quantité importante de cours ont été développés selon des principes et dans des environnements d'enseignement assisté par ordinateur (EAO). Nous montrons dans cet article comment utiliser un environnement d'EAO (NATAL) à partir d'un système EIAO (ART).

Après avoir fourni les caractéristiques du système qui effectue la traduction, nous montrons comment un programme initial écrit en NATAL est transformé en un programme équivalent écrit en ART. Un mode local permet d'utiliser des activités NATAL aussi bien que des activités ART.

Les méthodes d'interaction avec l'étudiant étant généralement plus riche au niveau des systèmes d'EIAO nous indiquons certains principes et un mécanisme intelligent capable de détecter les conditions où le mode EAO local nécessite une analyse des actions de l'étudiant au niveau EIAO qui pourra alors y suppléer.

THE USE OF CAL SYSTEMS IN ICAL SYSTEMS

Intelligent computer-assisted learning (ICAL) is still in the research stage, while a substantial number of courses have been developed in accordance with principles of and in contexts of computer-assisted learning (CAL). In this article, we will show how a CAL environment (NATAL) is used with an ICAL system (ART).

After describing the characteristics of the system that performs the translation, we will show how an initial program written in NATAL is transformed into an equivalent program written in ART. A local mode allows the use of NATAL activities as well as ART activities.

Since the methods of interaction with the student are generally more abundant in ICAL systems, we will indicate certain principles and an intelligent mechanism able to detect the conditions in which the local CAL mode requires an analysis of the student's actions in ICAL, which can then provide what is needed.

Utilisation de systèmes d'EAO dans des systèmes d'EIAO

Samy Bengio Claude Frasson Jan Gecsei

Université de Montréal, Dépt. IRO, C.P. 6128, succ. A,
Montréal (Québec), H3C 3J7 Canada

1. Introduction

Afin d'imiter, dans la mesure du possible, les enseignants humains, les systèmes tutoriels intelligents (STI) combinent les éléments d'EAO classique avec des techniques d'intelligence artificielle, dont les systèmes experts [Wate86]. Ces derniers se distinguent par leur capacité à comprendre la matière à enseigner et les difficultés de compréhension de l'élève. Ainsi ils peuvent aisément adapter leur enseignement aux objectifs et situations pédagogiques variées.

Dans les STI, la "compréhension" est obtenue par le fait que les informations sur la matière, sur l'état de l'étudiant et sur les méthodes pédagogiques sont emmagasinées dans des banques de connaissances (et non comme des algorithmes et données fixes). Ceci permet une manipulation relativement aisée de ces informations et une certaine flexibilité dans la présentation et vérification des sujets enseignés.
Cependant il est clair que la distinction entre EAO et STI est plutôt une matière de degré qu'une distinction absolue, et qu'un système d'enseignement informatisé ne devient pas automatiquement "intelligent" (même dans le sens figuratif) parce qu'il est implanté avec des techniques de l'IA. L'emploi d'un langage spécialisé ou d'une coquille pour le développement de systèmes experts n'est pas une garantie d'un résultat intelligent; ces moyens facilitent seulement la transmission de l'expertise humaine (particulièrement difficile à saisir en pédagogie) vers la machine.

Tandis que la plupart de la recherche est à présent concentrée sur le domaine des STI, les systèmes d'EAO sont toujours considérés comme adéquat dans un grand nombre de situations. De plus, des quantités considérables de didacticiels de type EAO ont été accumulés; néanmoins ces matériaux ne sont pas directement utilisables dans les STI. Une méthode qui permetterait d'intégrer ces didacticiels dans un STI est clairement désirable.

Une telle possibilité est l'un des objectifs du système Héron [Imbe88]. Héron est un champ d'expérimentation pour le développement de systèmes tutoriels éventuellement intelligents [Weng87] dans lesquels la représentation de l'expertise sur la matière, sur l'étudiant et sur la pédagogie, ainsi que la communication entre les blocs fonctionnels, sont uniformisés par l'utilisation du langage ART [Will84]. Ceci nous amènera à dégager une méthodologie commune pour une classe d'applications, qui pourra accélérer la conception des STI et disséminer ce savoir-faire rapidement aux endroits et pays où on en a le plus besoin.

La plupart des STI connus aujourd'hui traitent en profondeur de sujets étroits, tels que l'enseignement de parties choisies de la géométrie, ou d'un langage de programmation. Dans ces systèmes, le syllabus est développé en fonction de l'approche particulière utilisée; il n'est pas question de réutiliser les connaissances acquises dans un autre STI. Or, une telle possibilité pourrait mener à des grandes économies. Comme nous l'avons déjà mentionné, un autre objectif de notre projet est d'intégrer au système, autant que possible, les matériaux informatisés existant sur un sujet donné (cours, exposés, exemples, problèmes) dans un vaste "repositoire" appelé *curriculum* . Les éléments du curriculum peuvent être ensuite combinés pour créer des cours entiers, individualisés selon les buts pédagogiques et le niveau des élèves. Nous sommes convaincus qu'une telle démarche serait effective, même au prix d'une certaine incohérence entre les éléments rassemblés dans le curriculum. Ce type de STI devra posséder, en plus de l'expertise trouvée dans les STI courants, un nouveau type de connaissances concernant la *gestion* de curriculums et la *composition de cours* entiers.

Dans la suite de cet article nous exposons une des approches possibles à ce problème, consistant à traduire automatiquement les programmes écrits dans un langage auteur traditionnel (NATAL), au langage

ART. Au coeur de cette traduction est l'emploi d'un traducteur et d'un interprète qui ensemble transforment un programme quelconque écrit en NATAL en un programme équivalent en ART. La section suivante décrit brièvement les composantes de l'architecture alors que la section 3 esquisse les principes de la solution et illumine certains détails par le biais d'un exemple simple mais complet. Ceci sera d'un intérêt méthodologique pour ceux qui développent des ITS en utilisant une coquille pour système expert.

2. Les composants de la solution

2.1 NATAL

NATAL [Brah84] est un langage auteur largement utilisé au Canada. C'est un langage procédural simple à utiliser et spécialisé dans les applications pédagogiques. Les blocs d'enseignement sont séparés du contrôle du séquencement. De plus, NATAL possède un système de fichiers puissant et un sous-système graphique.

2.2 ART

Automatic Reasoning Tool est le logiciel principal utilisé par le projet Héron. Il incorpore de nombreux outils pour le traitement des connaissances déclaratives (faits, schémas), des connaissances procédurales (règles), ainsi qu'un moteur d'inférence à chaînage avant et arrière, la programmation orientée objet et un sous-système graphique. ART permet les appels à des fonctions LISP écrites par l'usager.

3. Architecture du système proposé

3.1 Généralités

L'architecture que nous proposons se trouve schématisée à la figure 1. On y retrouve un programme écrit en NATAL, qui pourrait être exécuté dans un environnement NATAL (comme un compilateur par exemple). Ce programme, pour être intégré à ART, est tout d'abord traduit en une *structure ART équivalente* qui, à l'aide d'un interpréteur, devient un *programme ART équivalent*, qui simule exactement le comportement du programme NATAL original.Ce programme est ensuite intégré à un curriculum de STI avec le module *intégrateur*.

Nous allons voir en détails comment ces différents modules fonctionnent, et les illustrer à l'aide d'un exemple sur l'enseignement de la grammaire française.

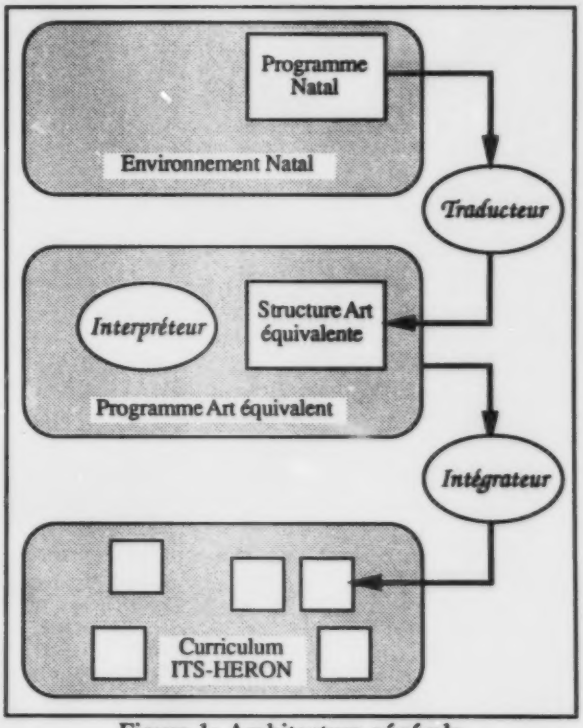


Figure 1: Architecture générale

3.2 Le traducteur

Le traducteur a pour but de traduire n'importe quel programme NATAL, et de créer une structure ART équivalente. Pour ce faire, nous avons d'abord établi la grammaire exacte du langage NATAL, c'est-à-dire la définition et la syntaxe de toutes les commandes NATAL.

Nous avons également établi la syntaxe de la structure ART souhaitée. Cette structure est en fait une série de schémas ART qui représentent chacun une commande NATAL. Si un programme NATAL se compose de dix commandes, il y aura en sortie dix schémas ART, un pour chaque commande. Enfin, la traduction se fait à l'aide d'un système d'écriture de compilateurs, le SEC [Boch78].

La figure 2 nous donne un exemple de programme NATAL que l'on pourrait traduire avec le traducteur.

Programme Natal

```
 1  test: ENTRY PROCEDURE;
 2    $PROMPT <- 'Question ';
 3    CALL *question;
 4    IF (RIGHT = TRUE) THEN
 5      CALL *print('Felicitations!');
 6  END test;

 7  *question: UNIT;
 8    DISPLAY &E &L5 Question No 1
        &L2 Ecrire le verbe etre,
        &L a la 2eme personne du
        &L singulier, passe simple &;
 9    RESPONSE;
10    EDIT SHDN, PUNC, MULSP;
11    RIGHT CC("etais");
12  END *question;

13  *print: UNIT(STRING x);
14    DISPLAY &E &L5 &V(x) &;
15  END *print;
```

Figure 2: Un programme NATAL

Nous avons numéroté les commandes pour plus de clarté. Le programme pose une question à l'usager, lui demandant d'écrire le verbe *être*, à la deuxième personne du singulier, au passé simple. Si l'étudiant répond correctement, le système lui retournera le message "Félicitations!".

Ce programme NATAL est donc fourni au traducteur, qui créé un schéma pour chaque commande (dans notre cas, quinze). On peut voir à la figure 3 une partie du résultat obtenu.

Partie de la structure Art équivalente

```
(defschema natal-prog-4
  (instance-of commande-if)
  (commande-then natal-prog-5)
  (commande-else vide)
  (condition (= RIGHT TRUE))
)

(defschema natal-prog-5
  (instance-of commande-call)
  (schema-du-bloc-corresp natal-prog-13)
  (parametres ("Felicitations!" ))
)

(defschema natal-prog-13
  (instance-of commande-unit)
  (nom-de-unit *print)
  (parametres ((STRING x)))
)
```

Figure 3: Structure ART équivalente

Ici, les 3 schémas correspondent respectivement aux commandes numéro 4 (le IF), 5 (le CALL), et 13 (le UNIT). On peut voir, par exemple, que le schéma correspondant au IF contient toutes les informations nécessaires à l'exécution d'un IF, c'est-a-dire le type de la commande (commande-if), la condition à vérifier (que RIGHT soit égal à TRUE), le schéma de la commande à exécuter si la condition s'avère vraie (natal-prog-5), et le schéma de la commande à exécuter si la condition s'avère fausse (vide, c'est-à-dire qu'il n'y en a pas).

3.3 L'interpréteur

L'interpréteur a été conçu avec les principes de la programmation orientée objet [Bail87], dont ART fourni les principaux outils.

En programmation orientée objet, la notion centrale est l'*objet*, qui représente généralement des entités traitées par le programme. On peut leur donner des caractéristiques, et on peut grouper les objets qui se ressemblent. On parle alors de *classes* d'objets, qui sont des ensembles dont les éléments sont des *instances* de cette classe. Pour accéder à ces objets, on doit créer des *méthodes* pour chaque objet, ou pour chaque classe d'objets. Les méthodes sont les procédures qui peuvent manipuler les objets. Une méthode définie pour une classe est accessible à tous les objets de cette classe, sauf ceux pour qui cette même méthode à été redéfinie spécifiquement.

En ART, les objets sont représentés par des *schémas*, et les méthodes par des *defactions*. La figure 4 montre un exemple de méthode.

Méthode pour la commande Call

```
(defaction exec-methode (commande-call) ()
  (empile-pointeur
    (+ 1 (schema-vers-num commande-call)))
  (empile-parametres
    (get-schema-value commande-call 'parametres))
  (get-schema-value commande-call
    'schema-du-bloc-corresp)
)
```

Figure 4: Méthode de la commande CALL

Cette méthode s'applique à tous les objets de la classe *commande-call*, c'est-à-dire toutes les commandes CALL d'un programme NATAL. Le contenu de la méthode est écrit en LISP, et peut se résumer ainsi: empiler le pointeur de la prochaine commande à être exécutée normalement, empiler la liste des paramètres contenus dans la commande CALL, et retourner le nom du schéma correspondant à l'unité appelée lors du CALL.

Ces méthodes sont appelées dans un ordre bien déterminé, qui est géré par le *moteur de l'interprète*, composé d'une série de fonctions écrites en LISP. On peut en voir quelques-unes à l'aide de la figure 5.

Moteur de l'interprète

```
(defun interprete (schema-pere)
  (prepare-variables-globales)
  (prepare-moteur schema-pere)
  (exec-liste-commandes (num-vers-schema 1))
)

(defun exec-liste-commandes (nom-de-schema)
  (let* ((schema-suivant
          (invoke 'exec-methode nom-de-schema)))
    (modifie-moteur-interprete
      'schema-actuel schema-suivant)
    (when (not (null schema-suivant))
      (exec-liste-commandes schema-suivant))
    )
)
```

Figure 5: Le Moteur de l'interprète

On remarque la définition de la fonction *interprète* qui s'occupe de préparer les variables globales de l'environnement NATAL, de préparer les variables servant au moteur de l'interprète (comme le pointeur actuel, la pile des retours d'appels, la pile des paramètres, etc), et de lancer l'execution du programme en commençant par la première commande.

La fonction *exec-liste-commandes*, récursive, s'occupe d'exécuter une première commande à l'aide de sa méthode correspondante, et se rappelle avec la prochaine commande à exécuter, jusqu'à ce que le programme NATAL soit terminé.

3.4 L'intégration

L'intégration de programmes NATAL dans des ITS se fait au niveau du curriculum de ces ITS. C'est le curriculum qui contient la matière à enseigner, organisée en une série d'*unités-matière* qui contiennent chacune l'information relative à l'enseignement d'un élément de matière.

Le module d'intégration permet d'insérer, en tout ou en partie, un programme NATAL dans une unité-matière, déjà existante, ou nouvellement créée. Il doit toutefois utiliser l'aide humaine pour caractériser la nouvelle unité-matière. En effet, c'est l'enseignant humain qui détermine le niveau de l'unité, ses pré-requis, son sujet, et toute information nécessaire à l'intégration exacte de cette unité-matière dans le curriculum.

4. Conclusion

Dans cet article, nous avons montré une méthode permettant d'insérer dans un STI le matériel développé pour des systèmes d'EAO conventionnels et ainsi accélérer le développement des STI. Le traducteur et l'interpréteur sont complets; il reste à faire des expériences étendues pour vérifier l'efficacité et l'utilité de cette approche dans le contexte d'un large curriculum.

5. Bibliographie

[Bail87] Bailly, C., Challine, J.F., Gloess, P.Y., Ferri, H.C., Marchesin, B. (1987) *Les langages orientés objets*, Cépadues Éditions.

[Boch78] Bochmann, G.V., Ward, P. (1978) Compiler writing system for attribute grammars, *Computer Journal*, 21:2, 144-148.

[Brah84] Brahan, J. et Godfrey, D. (1984) *Computer-aided learning using the NATAL langage*, Softwords, Press Porcépic.

[Imbe88] Imbeau, G., Frasson, C. et Gauthier G. (1988) A Multi-Expert System for Large Scale Intelligent Tutoring, dans *ITS-88*, ACM.

[Wate86] Waterman, D. A. (1986) *A Guide to Expert Systems*, Addison-Wesley, Reading, MA.

[Weng87] Wenger, E. (1987) *ARTificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to Communication of Knowledge*, Morgan Kaufmann Publishers Inc.

[Will84] Williams, C. (1984) ART the advanced reasoning tools - conceptual overview, Inference Corporation.

INNOVATIONS IN COURSEWARE DEVELOPMENT

A. Le Xuan
Alphamega Knowledge Engineering
Montreal, PQ

R. Shinghal
Computer Science Department
Concordia University
Montreal, PQ

Creating an Intelligent Computer-Assisted-Instruction System that can communicate knowledge at the appropriate level to a student selectively highlighting the interrelationships and ramifications of items, recalling information, probing understanding, explaining different areas in more depth, and skipping over seemingly known materials, is a task requiring several operations. But the most important operation is the generation of the knowledge base before any of the other operations can be performed. This paper describes the methods and techniques called Semantic Analysis, a rigorous procedure to generate all the subordinate and related concepts of the domain we wish to teach.

INNOVATIONS DANS LE DÉVELOPPEMENT DES DIDACTICIELS

La création d'un système intelligent d'enseignement assisté par ordinateur intelligent pouvant transmettre des connaissances au niveau approprié à un élève en soulignant les relations et les ramifications des éléments, en retraçant des informations, en évaluant la compréhension, en fournissant des explications plus approfondies et en sautant des sujets connus, est une tâche comportant plusieurs opérations dont la plus importante est la génération de la base des connaissances indispensables à l'exécution des autres opérations. Cette communication décrit les méthodes et techniques de l'analyse sémantique, laquelle est une procédure rigoureuse de génération des concepts subordonnés et connexes dans la matière à enseigner.

INNOVATIONS IN SOFTWARE DEVELOPMENT

Albert Le Xuan
Alphamega Knowledge Engineering, 4987 Walkley Ave, Montreal, Que H4V 2M3

And

Rajjan Shinghal
Concordia University, Computer Science Department, 1455 De Maisonneuve W.
Montreal, Que H3G 1M8, Canada

1.0 ABSTRACT

Creating a knowledge base for a Computer-Assisted-Instruction system that can communicate knowledge at the appropriate level to a student selectively highlighting the interrelationships and ramifications of items, recalling information, probing, understanding, explaining different areas in more depth, and skipping over seemingly known materials, is a task requiring at least the following operations:

1. Automating instruction in the manner that is easy to understand,
2. Simulating problem-solving process to enable the program to evaluate practical solutions and offer assistance when the student is uncertain what to do next,
3. Anticipating and coding responses for all possible student solution paths,
4. Encoding teaching knowledge as "dialogue management" rules,
5. Applying expert system techniques to teaching decision-making, and
6. Generating - without involuntary repetitions or omissions - the knowledge base before any of the above operations can be performed.

This paper describes the methods and techniques called Semantic Analysis, a rigorous procedure to generate all the subordinate and related concepts of a domain we wish to teach.

2.0 INTRODUCTION

In this paper we present the notion of Semantic Analysis (SA) and how it can be used to produce the knowledge base for the development of the courseware. We explain some of the pre-requisites and basic theories behind SA. We discuss what is meant by learning and teaching a concept, and how a complex concept can be taught in te terms of simpler or contributory concepts. We emphasize the importance of examples and non-examples in learning and teaching a concept, and we then show by examples how a semantic analysis is performed to generate the knowledge base.

3.0 THE PROBLEM OF DEVELOPING HIGH-QUALITY INSTRUCTIONAL MATERIALS

3.1 A Great Composition: Professor Jerrold Zacharias suggested in a 1956 memorandum which lead to the creation of the Physical Science Study Commission (PSSC) that, "Education is like hi-fi. To have a good hi-fi, you need a good performer on a good day, a good recording, a good pressing of it, a good pick-up and amplifier; you need a room with good acoustical qualities and a person with the intent to listen. But most important of all is the composition itself; without a great composition itself; without a great composition, everything else is pointless." (Silberman, 1970)

3.2 Cooperation Between Knowledge Engineers & Domain Experts: The planning of a good learning system development project ideally provides for close cooperation between knowledge engineers (KEs) & domain experts (DEs). The DE usually holds a higher position than the level for which the system is to be built. He (for brevity, we shall use masculine pronouns for both genders) should know the domain of his field thoroughly as well as the problems encountered at various levels within the specialization. The KE applies learning theories to the practical task of learning and teaching. Ideally he should have acquired the methods & techniques of SA.

3.3 Allocation of Responsibilities: Although a course can be developed by one KE working with one DE, or by one person acting as both, it is better to have one KE working with several DEs, programmers, artists and editor to reduce the costs.

3.4 Sets of Steps: While designing a learning system, the KE and the DE go through the following steps:

1) Determining the needs
2) Identifying the problem
3) Specifying the target population
4) Specifying the entering behavior
5) Generating the knowledge base (KB)
6) Converting the KB into story-boards
7) Testing and revising until the material meets specifications
8) Entering the material into the computer and so on.

Some textbooks explain quite clearly how to perform all the steps except Step Six: "Generating the knowledge base". It is the raw material with which we build the learning system. The success of a learning system depends on the quality of the knowledge base, and it is the SA that helps us to generate the knowledge base.

4.0 SOME THEORIES BEHIND THE SA

4.1 Introduction: The term Semantic Analysis may be new, but the idea of SA is not. Every good teacher or educator has been using it in some form or other without realizing that it is called SA.

4.2 Some Examples: In his Discourse on the Method (1637), René Descartes was certainly using an SA technique when he wrote about dividing each problem into as many parts as feasible and requisite for the solution of a problem. Unfortunately Descartes never showed us exactly how to go about such division.

According to Skemp (1971),"lower-order concepts must be present before the next stage of abstraction is possible. In building up the structure of successive abstractions, if a particluar level is imperfectly understood, everything from there on is in peril. This dependency is probably greater in mathematics than in any other."

Robert Gagné (1985) maintains that readiness to learn is essentially a function of pre-requisite learning. "Ensure that he has acquired the pre-requisite capabilities and he will be able to learn. When he is capable of tasks d & e, he is ready to learn b; when he is capable of tasks f & g, he is ready to learn c; and when he is capable of taks b and c, he is ready to learn a", which is the most abstract concept in the hierarchy, as shown in Figure 1.

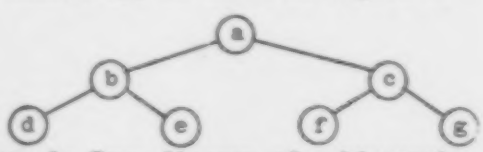


Figure 1: Tree diagram of a hierarchy

David Ausubel (1960), who is in agreement with Gagné on this point, has stated it eloquently thus: "If I had to reduce all of educational psychology in one principle, I would say this: 'The most important single factor influencing learning is what the learner already knows. Ascertain this and teach him according.'"

5.0 SEMANTIC ANALYSIS

In practice, what Skemp, Gagné and Ausubel say amounts to this: before one can understand and learn a new concept, one has to discover and learn its subordinates and related rules, and so on, until one reaches those one already knows. Unfortunately, Skemp, Gagné and Ausubel have not shown us how to discover the subordinates of a concept - at least in a concrete, simple and systematic way. Le Xuan and Chassain (1975, 1985) have developed a method and technique which allows us to generate - without involuntary omissions or repetitions - all the necessary subordinate concepts and rules of any concept rule, principle or law. It is called behavior analysis and topdown conceptual analysis, which is one form of semantic analysis (SA). Doing an SA is like decomposing a composite number into its prime factors. If one has to factor the number 24 for instance, one will always obtain the prime factors 2, 2, 2, 3, no matter how one decomposes it (see Figures 2,3).

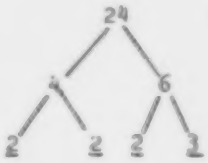


Figure 2

or

Figure 3

6.0 USING SA TO GENERATE KNOWLEDGE BASES

The following is the SA the authors have performed to generate the knowledge base for the development of the learning system: WHEN CAN WE SAY THAT AN INDIVIDUAL HAS LEARNED A CONCEPT? This SA was performed from a question, but it can also be performed from any other stimulus (input): A term, symbol, photograph, object, sentence, or text. The KE and the DE need not know the answers to all the questions but they should know where to find the answers to all the questions with the help of textbooks, related documents and encyclopedias.

5A : WHEN CAN WE SAY THAT AN INDIVIDUAL HAS LEARNED A CONCEPT?

Part A : Input-output Inventory

Input

1.0 When can we say that an individual has learned a concept?

2.0 Answer: When an ... a concept.

3.0 Example: Consider ...

4.0 Non-example: When ... dog.

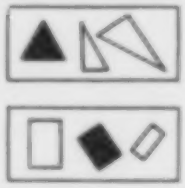
5.0 Generalizing ... classes

Output

2.0 Answer: When an individual is capable of generalizing within a class and of discriminating between that class and other classes, we can say that he has learned a concept.

3.0 Example: Consider the following two classes of objects:

Generalization within classes



Discrimination among classes

When an individual makes the same response "triangle" to each of the objects of class (1), and the same response "rectangle" to each member of class (2), he is generalizing within the classes of triangles and rectangles; he is also discriminating between the class of triangles and the class of rectangles. The psychologist would say that the individual has learned the concepts of "triangles" and "rectangles".

4.0 Non-example: When an individual can recognize a dog but is unable to distinguish a dog from a fox, we say that he has not learned the concept of "dog".

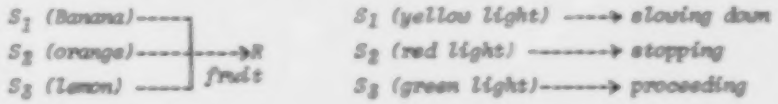
5.0 Generalizing within classes and discriminating between classes

PR Short for Pre-Requisite, for which we mean that the content of this item (example) contains no terms (or concepts) calling for further analysis (breakdown).

PR

6.0 Definitions: Making the same response to different stimuli (objects, qualities, quantities, events, relations) is generalizing within classes and making different responses to different stimuli is discriminating between classes.

7.0 Examples & Non-examples: Saying "fruit" in the presence of bananas, oranges and lemons is generalizing within a class of fruits. Slowing down when the traffic light is yellow, stopping when the light is red, and proceeding when the light is green is an example of discrimination.



Generalization within a class Discrimination between classes

Saying "horse" in the presence of black horses, white horses, old horses, young horses, all kinds of horses, is generalizing within a class of horses.

Saying "fox" in the presence of foxes and saying "dog" in the presence of greyhounds, setters, pointers is discriminating between a class of foxes and a class dogs.

PR

SA : WHEN ... CONCEPT
Part B : Representation of the SA

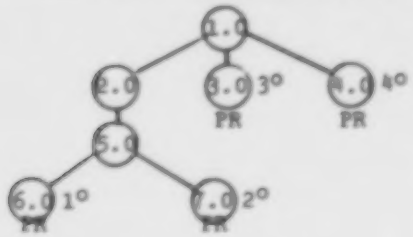


Figure 2: Tree diagram representing a semantic network where each node is a component part & each arrow or arc, a link connecting the parts.

Note: The usefulness of the tree diagram would be more apparent if the domain is more complex.

SA : WHEN ... CONCEPT
Part C : Sequential Index

1° (6.0-5.0-2.0) Definitions of generalization & discrimination

2° (7.0-5.0-2.0) Examples & Non-examples

3° (3.0-1.0) Example of concept acquisition

4° (4.0-1.0) Non-example of concept acquisition

7.0 RECAPITULATION

7.1 The quality of the software: The quality of the software depends on the quality of the raw material with which it is built : the knowledge base which is the end-product of the semantic analysis: the semantic network. In other words, semantic analysis is the tool which allows us to produce high quality knowledge base.

7.2 How was Part A generated: Each input is written on the left side of the page. It generates one or more outputs which appear(s) on the right side of the page. Each of these inputs will in turn become a new input for further outputs and so on, until one reaches outputs which call for no further analysis: these are the prerequsites or knowledge which is already in the repertoire of the target audience.

7.3 How was Part B generated: The reorganization of the input-output inventory into a semantic network, we drew a tree diagram where each node represents a component part of the network and the arcs or arrows represent the relationship among the parts of the network.

7.4 How was Part C generated: To generate Part C we proceeded as follows:

1) We picked out Item 1° on the tree diagram, then looked at the input-output inventory, and copied the content of 1° on a separate sheet of paper. This is the first element of network.

2) Next, we picked out Item 2° and copied its content. We proceeded in the same manner until the last item had been disposed of. The end-result is the sequential index or knowledge base we seek to develop. It is the raw material we use to develop learning systems.

3) Note: The analysis starts from the top downwards and the presentation of the learning system starts from the bottom upwards.

7.5 The use of SA is an innovation: To our knowledge, Le Xuan & Chassain (1975, 1985) are the only authors who have shown us - at least in a concrete and systematic way - how to break down a concept or task into its component parts without involuntary omissions or repetitions.

8.0 REFERENCES & BOBLIOGRAPHY

1. Ausubel, D.P. (1960). The Uses of Advance Organizers in the Learning of Meaningful Verbal Material. Journal of Educational Psychology, 1960, 51, pp. 267-72.

2. Descartes, René (1637). Discourse on the Methods. London: Nelson's University Press.

3. Gagné, Robert (1985). Conditions of Learning. New York: Holt, Rinehart & Winston.

4. Le Xuan & Chassain (1975). Analyse Comportementale: Paris: F. Nathan.

5. _____ (1985). Semantic Analysis. Montreal: Pedagogical & Technological Innovations.

6. Silberman, Charles E. (1970). Crisis in the Classroom. New York: Random House.

7. Skemp, R.R. (1971). The Psychology of Learning Mathematics. London: Penguin Books.

Final Note: We need more than four pages to show how powerful a tool SA is.

TUTEURS INTELLIGENTS VS SYSTÈMES PÉDAGOGIQUES À BASE DE CONNAISSANCES ERRONÉES: COMMENT UTILISER LES ERREURS PRODUITES PAR UN APPRENANT

B. Dumont
Dép. technologies nouvelles et enseignement
Institut national de recherche pédagogique
Paris, France

Beaucoup de travaux plus ou moins récents, depuis WEST et BUGGY, sont actuellement classés sous la rubrique "tutoriels intelligents". À partir d'exemples je tenterai de montrer ce que ces logiciels savent fair et en quoi ils sont encore très loin de jouer un réel rôle de tuteur. À partir de là je poserai deux questions me paraissant fondamentales.

1) A-t-on des chances de voir, plus ou moins prochainement, apparaître de 'réels' tutoriels (intelligents ou non) qui ne soient pas simplement des logiciels très sophistiqués, du point de vue informatique, mais pas beaucoup plus avancés, du point de vue pédagogique, que leurs ancêtres issus de l'EAO?

2) A-t-on vraiment besoin de logiciels de type "tutoriel"? Si oui, pour quel public: formation initiale ou formation continue?

A partir d'un logiciel développé à l'INRP, appelé FRACT, à partir d'une base de connaissances sur le calcul des fractions, et dans le cadre d'une "pédagogie de l'erreur", je montrerai quels avantages l'on peut tirer d'un Système à Base de Connaissances d'Erreurs, aussi bien pour les enseignants que pour les apprenants.

INTELLIGENT TUTORING SYSTEMS VS PEDAGOGICAL ERRONEOUS KNOWLEDGE BASED SYSTEMS: HOW TO USE LEARNERS' ERRORS

Many fairly recent projects, since WEST and BUGGY, are currently categorized as "intelligent tutoring systems". Using examples, I will try to show what these software packages can do and how they are still very far from playing a real tutoring role. On the basis of this, I will ask two questions that seem fundamental to me.

1) Is there any chance in the foreseeable future of our having 'real' tutoring systems (whether intelligent or not) which will be more than mere software packages (highly sophisticated ones, from a data processing point of view) representing little educational progress from their CAL ancestors?

2) Do we really need software of the "tutorial" type? If so, for which clientele: those in initial schooling or those in continuing education?

Using a software program developed at INRP, called FRACT, and a knowledge base for the calculation of fractions, and following an educational approach based on errors, I will demonstrate the advantages to be derived by both teachers and learners from an "erroneous knowledge based system".

TUTEURS INTELLIGENTS VS SYSTEMES PEDAGOGIQUES A BASE DE CONNAISSANCES ERRONEES : COMMENT UTILISER LES ERREURS PRODUITES PAR UN APPRENANT

Bernard DUMONT
Département "Technologies Nouvelles et Enseignement"

INSTITUT NATIONAL DE RECHERCHE PEDAGOGIQUE
29, rue d'Ulm
F 75230 PARIS CEDEX 05

I- POUR UNE PEDAGOGIE DE L'ERREUR

Dans la vision classique de l'enseignement, le maître est chargé, par la communauté éducative, de transmettre la "bonne parole", faire entrer dans la tête des élèves les savoirs, c'est-à-dire ce qui est "vrai", démontré, reconnu. C'est aussi lui qui reçoit la mission de dire si tel élève a appris (ou, plus difficile, compris) telle notion, s'il peut passer dans la classe supérieure. Pour l'élève la confusion survient souvent entre la "vérité" scientifique et la vérité du maître, "si le maître le dit, c'est vrai".

L'erreur reçoit donc, dans la situation scolaire ordinaire, un statut dévalorisant et culpabilisant pour celui ou celle qui la produit.

Il n'est donc pas inutile d'insister sur l'intérêt pédagogique des erreurs, tant en formation en face à face que pour l'enseignement à distance.

Dans le premier cas, le rapport de l'équipe de recherche Articulation école-collège de l'I.N.R.P. (1) prouve bien que le comportement des maîtres le plus répandu devant une production d'erreurs en mathématiques, est la rectification du discours afin que le maître énonce (ou fasse énoncer) la "bonne" réponse :

"- corriger une erreur c'est bien plus rechercher la bonne réponse qu'analyser, décortiquer cette erreur ou produire des arguments.

- les élèves sont sollicités presque exclusivement pour donner la bonne réponse.

- l'enseignant intervient soit pour émailler la correction de commentaires ne comportant pas d'éléments d'information, soit pour aider les élèves en rappelant une règle, un énoncé ou en donnant une explication. On peut d'ailleurs noter que cette explication vient souvent à la suite de la bonne réponse donnée par un élève ou par lui-même."

De l'étude ci-dessus citée, il ressort qu'il y a donc peu ou pas de récupération de l'erreur produite par l'élève. Celui-ci est donc renvoyé à la réponse de référence ; il n'a appris qu'une seule chose : il avait "faux". Or, dans l'optique de ce que j'appelle une pédagogie de l'erreur, celle-ci peut être une source d'information pour le maître.

Cette pédagogie repose sur l'idée simple suivante : on ne possède pas bien un domaine de connaissances uniquement en étant capable d'énoncer des règles justes mais surtout en ayant acquis les savoir-faire permettant d'utiliser ces connaissances et en ayant compris tout ce que l'on ne peut pas appliquer, par exemple, les raisons pour lesquelles telle propriété ne peut être étendue à telle situation.

COMMENT FAVORISER UNE TELLE PEDAGOGIE ?

Du point de vue de la formation des maîtres, en tout premier lieu, il serait nécessaire de les confronter aux erreurs : qu'ils soient capables de les reconnaître et, surtout, de les comprendre pour s'en servir en vue d'activités de remédiation. Le logiciel BUGGY peut favoriser cette approche, et il pourrait aisément en être de même de FRACT-1 que je présente ci-dessous. Dans ce type d'activité l'élève-maître pose des questions à la machine; celle-ci, se comportant en mauvais élève (mais de manière systématique, ce qui ne semble pas être le cas de la majorité des élèves), c'est-à-dire en combinant des règles vraies et des règles fausses, répond de manière incorrecte; l'élève-maître, seul ou après discussion avec ses pairs, doit reconstruire des cheminements erronés possibles ayant conduit de l'énoncé au résultat faux.

Cela suppose des connaissances didactiques précises sur ces phénomènes d'erreur, sur les conditions d'apparition, les facteurs favorisant tel ou tel comportement, les enchaînements d'erreurs.

DES OUTILS POUR L'ENSEIGNEMENT

Une pédagogie de l'erreur conduit aussi à utiliser des exercices très particuliers, des exercices incitatifs à l'erreur.

Dans la perspective de l'enseignement à distance ou de l'apprentissage individualisé, une "pédagogie de l'erreur" peut être mise en pratique et donner lieu en particulier à des tests d'auto-évaluation utiles et efficaces.

Dans le cas d'un apprentissage sans face-à-face, le temps s'écoulant entre la production d'une erreur et la correction peut être très long (plusieurs semaines pour la plupart des cours par correspondance), l'erreur, si elle est profonde, est donc déjà bien ancrée dans la mémoire de l'apprenant lorsque l'éventuel remède est apporté, ce qui diminue ipso facto son effet. On peut donc, pour être doublement efficace, proposer des exercices auto-évaluatifs/auto-correctifs ayant pour buts :

- d'être incitatifs aux erreurs
- de repérer les erreurs "intéressantes"
- d'offrir des moyens de faire comprendre à l'apprenant les causes de ses erreurs etd'augmenter sa connaissance du domaine étudié.

II. TUTORIELS INTELLIGENTS VS SYSTEMES A BASE DE CONNAISSANCES

L'article de Mark R. LEPPER et de Ruth W. CHABAY : "Socializing the Intelligent Tutor : Bringing Empathy to Computer" (2) me semble bien situer la problématique de ce paragraphe.

Après avoir cité GOETHE ("En toute chose, nous n'apprenons que de ceux que nous aimons"), les auteurs se posent des questions sur les ordinateurs promus -Intelligence Artificielle oblige- au rang de tuteurs.

Leur première constatation est qu'il existe peu de recherches sur le tutorat "réel" (humain) et, dans le peu qui existe, les objets principaux d'étude sont reliés à des préoccupations d'ordre cognitif et non pas affectif.

La seconde est que, en dépit de cette quasi vacuité, les logiciels éducatifs "en pointe" comportent tous un module tutoriel même si, vus de près, il est difficile de savoir à partir de quelle information des décisions pédagogiques sont prises. Dans le meilleur des cas il semblerait que, comme dans le cas humain, ce soient des comportements de type cognitif qui entrent en jeu.

Les auteurs émettent alors l'hypothèse que, dans la formation de type face à face/enseigné-enseignant, et plus encore dans le cas d'un tutorat où l'enseignant est avec un seul apprenant, les facteurs affectifs sont primordiaux et que les motivations de cet apprenant sont réellement prises en compte par l'enseignant.

Les auteurs citent, à l'appui de leurs dires sur la pauvreté de ce que les concepteurs de logiciels appellent néanmoins des "tutoriels", un exemple de commentaire à partir d'une erreur dans une multiplication (3) : à la réponse 32 pour l'opération demandée : 87 x 43, le programme répondit :

"Your answer is wrong.
Possible causes of error :
1. You multiplied the number in the multiplicand by the number directly beneath it in the multiplier, and you wrote down the carried number, ignoring the units number".

Je partage l'inquiétude des auteurs de l'article qui se demandent sur quelle base ce "tuteur" a décidé d'envoyer un tel commentaire, comment un élève de 4ème année du primaire peut réagir à cette information et, en particulier, s'il aura envie de recommencer ...?

Ils se posent aussi d'intéressantes questions que je reprends volontiers à mon compte :

- quels principes et quelles stratégies de motivation peut-on ou doit-on introduire dans un ordinateur?
- comment la "personnalité" d'un ordinateur peut influencer les performances et les apprentissages d'un utilisateur?
- comment un enseignant choisit de réagir à la production d'une erreur par l'apprenant (attendre d'autres erreurs, réagir immédiatement, corriger, expliquer...)? comment adapter ce type de choix en fonction de la personnalité de l'apprenant ? comment faire jouer un tel rôle à un ordinateur?
- l'ordinateur pourra-t-il se mettre un jour "à la place" de l'apprenant? ("emphatetic computer").

Je garde une de leurs dernières remarques pour élargir le problème :
"A sophisticated computer tutor should also know to phrase comments and provide feedback so as to create and sustain student motivation (...). It should provide 'informational' specific feedback on performance rather than 'controlling' feedback, (...) for instance, and it should know how to present feedback in a fashion that maintains the learner's sense of competence and control" (2., p. 251).

Je n'ai pas repris toutes ces interrogations pour y apporter des réponses mais pour faire comprendre au lecteur en quoi ma démarche s'éloigne de celle des Tuteurs Intelligents.

Je retrouve dans beaucoup de projets sur les Tuteurs Intelligents des rêves de cette période qu'on aurait pu croire révolue et, mais l'un explique peut-être l'autre, des pionniers de l'E.A.O. reconvertis aux T.I.. Et puisque l'on cite en premier WEST comme un T.I. provenant d'une époque où n'existait que l'E.A.O., est-ce la preuve que la boucle est bouclée et que l'on a peu changé notre conception de l'utilisation de l'ordinateur dans la formation?

Il faudrait, je crois, commencer par préciser quel public est visé par les T.I. et quels sont les objectifs de formation visés.
Sont-ils destinés à une formation à distance ou à un complément de formation en face à face? L'utilisateur a-t-il d'autres moyens d'apprendre? A-t-il reçu une formation préalable à leur utilisation? Est-il un apprenant en formation initiale ou adulte en formation continue? Est-ce un cours ou une banque d'exercices? ...

Autant de questions que je me pose et qui me conduisent à m'inquiéter de ce renouveau de l'ordinateur-maître.

Pas plus que l'E.A.O., je ne crois les T.I. capables d'assurer la formation initiale et de "faire cours". Eventuellement, et dans des cas très particuliers et pour des publics adultes déjà formés, on peut imaginer que des T.I. pourront rendre service.

Ceci n'est pas en contradiction avec ma profonde conviction (fondée sur mon expérience d'enseignement à distance) qu'il est possible de créer des cours multimédias dans lesquels l'ordinateur peut jouer un rôle important et qui peuvent permettre, en dehors d'une formation de type face à face, à des adolescents ou à des adultes de se former.

Même très intelligents, ces tutoriels sont, de par leur conception, contraires à mes choix pédagogiques : ils recréent la situation maître-élève/détenteur de savoirs-récepteur de savoirs/celui-qui-sait-celui-qui-apprend avec, en moins cf. LEPPER & CHABAY) les avantages des relations humaines mais avec, en plus, l'intolérable Big Brother.

Pour moi l'ordinateur possède suffisamment de potentialités, encore peu utilisées dans la formation, pour que je n'investisse pas dans un

"tuteur", fût-il intelligent. Ma recherche va donc plus vers l'utilisation pédagogique de systèmes à bases de connaissances.

J'en vois de deux types non nécessairement disjoints : les bases de connaissances exactes et les bases de connaissances d'erreurs.

Du premier type je citerai CAMELIA, créé par Martial VIVET (4) et permettant de résoudre des problèmes d'intégration en particulier tout en étant capable de fournir des explications sur le choix des méthodes.

Le second type regroupe les bases de connaissancesd'erreurs. J'y place BUGGY et, bien entendu, FRACT. Elles peuvent être utilisables, comme je l'ai déjà dit, pour la formation des maîtres même sans interfaces très sophistiquées, à condition qu'elles aient été construites sur des informations réelles et qu'elles ne fassent pas intervenir des bugs irréalistes. Pour ce qui est de leur utilisation par des élèves, en formation initiale, je pense que la réflexion doit être poursuivie; la voie ouverte par LEPPER & CHABAY mérite attention, en particulier pour ce qui est des moyens de convaincre et de motiver.

III- PRESENTATION DE FRACT-1

L'idée de ce logiciel vient de BUGGY et de l'hypothèse suivant laquelle les comportements erronés dans les calculs élémentaires pouvaient suivre des "règles" proches des règles correctes.

Son rôle consiste à trouver toutes les combinaisons de règles vraies et fausses disponibles dans sa base de connaissances transformant une expression comportant une opération et deux fractions en une opération réduite à une seule fraction, simulant ainsi la situation dans laquelle la machine proposerait une addition (ou soustraction ou multiplication) de deux fractions (ou d'une fraction et d'un nombre) et récupérerait une réponse de l'apprenant.

FRACT-1 veut être une concrétisation de deux aspects possibles d'utilisation d'outils issus de l'Intelligence Artificielle dans l'enseignement, l'un pour les chercheurs en didactique, l'autre pour la formation des maîtres.

UN OUTIL POUR LA RECHERCHE EN DIDACTIQUE

Il s'agit de fournir aux didacticiens un logiciel permettant de traiter "intelligemment" un corpus d'erreurs répertoriées, c'est-à-dire capable de proposer divers chemins erronés pouvant rendre compte des productions d'élèves. Une telle aide se justifie dès que le travail "à la main" devient irréaliste, comme c'est le cas avec les fractions, parce que l'arbre des possibles est gigantesque et qu'il faudra recourir à des heuristiques de parcours.

Avec FRACT-1, notre banque d'erreurs est structurée en "îlots de connaissances", que les anglophones appellent "chunks", regroupant les règles correctes et incorrectes selon qu'elle appartiennent au domaine de la somme, de la différence, ou du produit de fractions; cette organisation n'est pas une partition puisque certaines règles peuvent appartenir à plusieurs îlots. A l'intérieur de chaque îlot, les règles sont identifiées suivant un niveau de fréquence d'utilisation.

UN OUTIL POUR LA FORMATION DES MAITRES

En s'inspirant encore de BUGGY, on peut estimer qu'un tel logiciel seraitutilisable avec profit dans le cadre de la formation des maîtres, comme je le disais plus haut, pour les mettre en situation de réfléchir sur descomportements erronés, en particulier sur des enchaînements de règles vraies ou fausses. L'étude même de certaines de ces règles, de la façon dont elles dérivent de règles vraies, est, à l'évidence, formatrice pour des enseignants. Une telle utilisation ne demande pas d'interfaces très sophistiquées et des étudiants ou des enseignants seraient tout à fait en mesure de travailler avec FRACT-1, aussi bien pour "lancer" le système en lui proposant des réponses que pour modifier, ajouter ousupprimer des règles, NEXPERT ™ et son environnement macIntosh ™ facilitant ce type d'accès.

PRESENTATION DE FRACT-2

Je ne suis pas en mesure de décrire ici le module "correction-remédiation" puisque, comme indiqué précedemment, notre réflexion se poursuit et notre recherche a besoin de temps pour mener à bien les expérimentations sur le terrain avec les différents protocoles envisagés. En revanche, je puis présenter maintenant divers éléments de réflexion sur les aspects pédagogiques de notre logiciel et sur les bases de sa conception.

L'erreur n'a pas besoin d'être systématique, au sens de répétable indéfiniment, pour mériter une attention particulière de la part du pédagogue et tenter de remédier à cette situation. En effet, si, par exemple, lors d'un exercice portant ostensiblement sur le calcul des fractions (et non pas s'il s'agit d'additionner deux fractions comme quinzième activité d'un problème de chimie), un apprenant additionne les numérateurs entre eux et les dénominateurs entre eux, c'est qu'il utilise une démarche erronée qu'il faut lui mettre en évidence, sur laquelle il faut l'amener à réfléchir; car, dans ces conditions "scolaires" de réponse, cette erreur peut être profonde et mériter remédiation. Il sera alors nécessaire d'essayer, avec un autre problème, de déterminer la profondeur "réelle" de cette erreur pour que la remédiation soit appropriée.

Néanmoins, si l'on admet que chaque élève dispose d'une très grande quantité de règles erronées, on comprend que l'on ait peu de chances de voir apparaître une quelconque stabilité dans ces comportements de réponses, même dans des contextes à première vue semblables. Notre seconde enquête nous a d'ailleurs fourni des exemples de copies où presque toutes les erreurs sont explicables par un modèle unique (cf. fig. 1) et d'autres où différents modèles erronés se côtoient (fig. 2).

Nous pensons qu'il ne faudra déclencher le module de remédiation que lorsqu'une erreur aura été décelée et que son niveau de profondeur aura été suffisamment bien estimé, sans attendre qu'à l'occasion

d'un grand nombre d'autres exercices l'apprenant reproduise la même erreur. En diminuant les phénomènes dus au temps (oubli des raisons de l'erreur, du cheminement réellement suivi, ...), l'explication devrait être plus efficace. FRACT-2 devra aussi mémoriser les erreurs produites pour s'assurer qu'après l'essai de remédiation les règles fausses ne seront plus utilisées.

Je souhaiterais que FRACT-2 ne se contente pas de "corriger" les erreurs repérées mais qu'il permette de construire un "vrai" modèle de l'élève, non pas pour le ramener sur le bon chemin en espérant qu'il sera capable de reproduire la "bonne" réponse attendue, mais un "modèle" pouvant rendre compte de la représentation que se fait l'apprenant du champ de connaissances considéré afinde lui apporter une aide efficace.

L'expertise pédagogique contenue dans FRACT-2 serait alors, d'une part, d'être capable de "profiter" des erreurs commises par l'apprenant pour construire un modèle et, d'autre part, d'être capable d'apporter à l'apprenant les connaissances manquantes et de "rectifier" ses connaissances erronées.

REFERENCES

1. Ouvrage collectif "Le statut de l'erreur dans l'enseignement en CM2 et en 6ème," Rapport de l'équipe de recherche Articulation Ecole-Collège I.N.R.P.-D.P. 1, Paris, 1987.
2. Ouvrage collectif, MANDL H., LESGOLD A. (Editors), learning Issues for Intelligent Tutoring Systems, Springer-Verlag, 1988.
3. ATTISHA M., YAZDANI M., An Expert System for diagnosing children's multiplication errors, Instructional Science, 13, 1984.
4. VIVET M., Expertise mathématique et informatique : CAMELIA, un logiciel pour raisonner et calculer, Thèse d'Etat Université Paris VI, 1984.

Pour plus de détails voir :

B. DUMONT, "Questionnements et interprétation des erreurs en mathématiques (élaboration de modèles pour la compréhension des comportements de réponse et la construction d'outils pédagogiques à supports technologiques)", Thèse de Doctorat d'Etat ès Sciences, Université Paris 7, 1989.

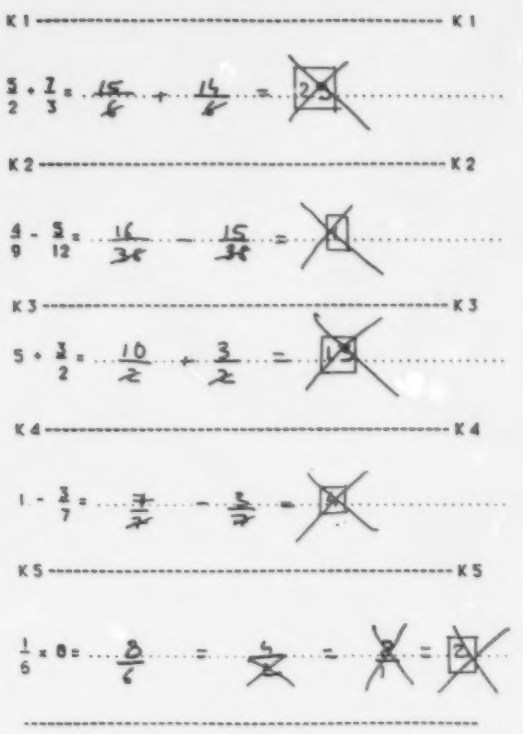


Fig. 1. Exemple de modèle unique

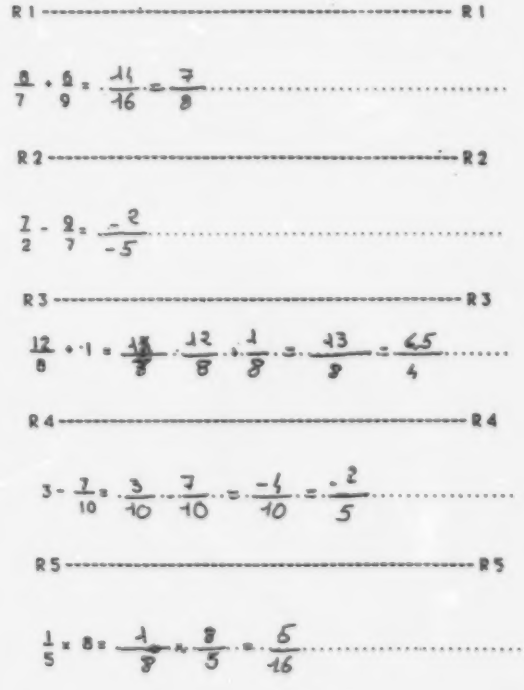


Fig. 2. Exemple de modèle double

SPÉCIFICATIONS D'UN SYSTÈME INFORMATIQUE SERVANT À LA CONCEPTION ET À LA RÉALISATION DE LOGICIELS ÉDUCATIFS À BASE DE CONNAISSANCES

M. Lavoie, M. Gagné, A. Jacques
Dép. de mathématiques et d'informatique
Université du Québec à Rimouski
Rimouski, PQ

Le présent article a pour but de présenter les spécifications d'un système informatique d'assistance à la conception et à la réalisation de logiciels éducatifs utilisant des bases de connaissances. Les spécifications du système permettent de planifier tout logiciel éducatif prenant en compte les objectifs, les phases et les produits de l'apprentissage. Un prototype de ce système informatique est fonctionnel. L'équipe de recherche s'en sert pour réaliser un véritable logiciel éducatif sur le thème de la logique séquentielle. À la lumière de cette expérimentation, les spécifications seront ajustées, et le système sera complété. Le système informatique découlant de ces spécifications pourra être couplé à un système-auteur ou un langage d'auteur pour former un système capable de produire des logiciels éducatifs à architecture souple et ouverte.

SPECIFICATIONS OF A COMPUTER SYSTEM FOR DESIGNING AND PRODUCING KNOWLEDGE-BASED INSTRUCTIONAL SOFTWARE

This paper presents the specifications of a software system that assists in the design and construction of knowledge-based instructional software. The features of this system allow the design of any instructional software that consider objectives, phases and results of the learning process. A prototype of this system is actually functional. The research team is using it to realize an instructional software on the theme of sequential logic. Taking into account this particular application, the specifications of the software system will be refined and the software will be completed. The software system ensuing from these specifications will allow the use of author languages and systems to produce instructional software with open and flexible architecture.

Spécifications d'un système informatique servant à la conception et à la réalisation de logiciels éducatifs à base de connaissances

Mario Lavoie, Michel Gagné, André Jacques

Département de mathématiques et d'informatique
Université du Québec à Rimouski
Rimouski, Qc

1- INTRODUCTION

La présente recherche s'inscrit dans le cadre d'un projet de recherche subventionné par le programme RDF de la DGERU[1] [Lavoie et al., 86].

Son objectif est d'élaborer et de réaliser un système informatique capable de produire des logiciels éducatifs assistant véritablement l'apprenant dans son processus d'apprentissage dans des domaines comme les mathématiques et les sciences. En fait, le système pourra s'appliquer à tout domaine où des hiérarchies d'apprentissage sont utilisables.

Sous sa forme finale, le système informatique permettra la conception et la réalisation de logiciels éducatifs permettant d'exhiber une partie importante des métaconcepts qui l'anime. Couplé avec un système-auteur ou un langage d'auteur classiques, il permettra à des enseignants de construire des logiciels éducatifs ayant une architecture souple et ouverte.

Plusieurs recherches se situent au niveau méta de l'E.I.A.O., c'est-à-dire au niveau de la conception et de la réalisation d'un système-auteur dans le contexte de l'E.I.A.O. Citons entre autres les projets MEDIAN[2] [Carton, 87], APT[3] [Fine, 80], ADAPT[4] [Mudrick, 84] et GAMETE[5] [Regourd, 87]. L'originalité de notre approche est qu'elle s'inscrit résolument dans le cadre d'une théorie de l'apprentissage. Selon la catégorisation de Wenger[6] [Wenger, 87], les logiciels produits à l'aide du système informatique projeté se situeront plus du côté des systèmes tutoraux fonctionnant selon un plan ou une stratégie que de ceux fonctionnant seulement de façon opportuniste, tout en prenant en compte cette dernière dimension.

Les spécifications du système informatique ont été élaborées. Elles s'inspirent d'un modèle didactique de logiciels éducatifs[7] [Lavoie, Beaulieu, 86]. Un prototype du système a été construit pour tester ces spécifications. L'équipe de recherche s'en sert pour réaliser un logiciel éducatif concernant la logique séquentielle que l'on enseigne aux élèves des ordres secondaire et collégiale de la province de Québec. A la lumière de cette expérimentation, les spécifications seront ajustées, et une version opératoire du système sera réalisée.

L'objectif du présent article consiste à exposer les spécifications de ce système d'assistance à la conception et à la réalisation de logiciels éducatifs utilisant des bases de connaissances.

2. PROBLÉMATIQUE

2.1 Problèmes caractérisant l'E.A.O.

Deux carences caractérisent les didacticiels classiques. Ce sont:
- une certaine rigidité de leur architecture et
- une insuffisance notoire des rétroactions qu'ils peuvent fournir à leur utilisateur.

La rigidité d'architecture des didacticiels classiques provient en partie de l'approche qui a été favorisée pour établir l'interface utilisateur-logiciel. L'interface en langage naturel n'étant pas disponible, les réalisateurs de didacticiels se sont tournés vers l'interface par menu. Cette approche, bien que amplement suffisante dans certaines situations, conduit à une architecture manquant parfois de souplesse, surtout dans le contexte des logiciels éducatifs.

Un deuxième facteur a contribué à roidir les articulations des didacticiels. Il s'agit des outils qui sont fournis aux réalisateurs de didacticiels. Des études sur différents systèmes-auteurs, tant européens qu'américains, le démontrent[8] [Barril et al., 85], [9] [De La Passardière, 85], [10] [Poc, 85], [11] [Le Calvez, 86].

En ce qui concerne la carence de rétroaction, rappelons que la majorité des didacticiels classiques ne possède même pas de fichier conservant la trace de ce que l'utilisateur a accompli avec le logiciel. Dans ces conditions, il est impossible au didacticiel de fournir la moindre rétroaction sur les apprentissages acquis et encore moins de fournir quelles que suggestions que ce soient concernant la suite des activités. Cette carence a été mise en évidence lors d'une étude concernant l'évaluation d'une application de l'enseignement assisté par ordinateur[12] [Brouard, 85]. A cette occasion, ce sont les utilisateurs du didacticiel qui ont déploré le manque de rétroaction fournie par celui-ci.

Même lorsque le didacticiel possède un fichier-trace, il ne lui est pas toujours possible, à

cause de la déficience de son architecture, d'interroger en temps réel ce fichier. L'opération se fait donc, dans le meilleur des cas, en différé.

2.2 Analyse de facteurs intervenant lors de l'élaboration de logiciels éducatifs

Plusieurs outils, découlant de techniques élaborées dans le contexte de la recherche en intelligence artificielle, sont présentement disponibles pour réaliser des logiciels éducatifs. Leur plus grand handicap s'avère être leur technicité. Ils ne sont pas accessibles à l'utilisateur courant de systèmes-auteurs. Ainsi, les personnes les plus aptes, par leur formation, à produire des didacticiels sont écartées de l'utilisation de ces outils puissants.

D'autre part, les personnes qui possèdent les techniques d'intelligence artificielle ne sont pas nécessairement celles qui ont des connaissances dans le domaine de l'apprentissage. Pour optimiser les chances de réussite de l'utilisation de ces techniques, il faut
- qu'elles s'appuyent sur une théorie de l'apprentissage qui a fait ses preuves et surtout
- qu'elles n'aillent pas à l'encontre de résultats psychopédagogiques déjà établis

La théorie de l'apprentissage que nous avons retenue pour encadrer nos travaux est la théorie de R.M. Gagné[13] [Gagné, Briggs, 79]. Cette théorie éclectique est suffisamment générale. Elle englobe entre autres certaines parties des théories de Gilbert, d'Ausubel et de Bruner[14] [Romiszowski, 81]. Dans le cadre de cette théorie, il est possible de prendre en compte le sujet traité, le niveau souhaité de compréhension de tel ou tel thème de la matière, ainsi que les moments où il est pertinent et intéressant d'exécuter certaines activités plutôt que telles autres. Ces variables sont respectivement contrôlées au moyen d'hiérarchies des objectifs d'apprentissage, par le biais des hiérarchies des produits de l'apprentissage et par le biais des phases de l'apprentissage. Les produits de l'apprentissage considérés par cette théorie s'échelonnent à partir de la discrimination, des concepts, des règles et des règles d'ordre supérieur, jusqu'aux stratégies cognitives nécessaires dans le contexte de la résolution de problèmes.

Une des approches prévalant dans le domaine de la recherche en intelligence artificielle consiste à atomiser la matière à faire apprendre pour l'articuler ensuite au moyen d'un quelconque moteur; par exemple, une base de règles dans le contexte d'un système expert. Or une étude a démontré [15] [Beeson, 77] qu'il est important, lorsque la matière à s'approprier recèle des habiletés intellectuelles, que celles-ci soient ordonnées en séquences hiérarchiques pour que leur apprentissage soit plus efficace. Cette situation est la règle plutôt que l'exception dans des domaines comme les sciences et les mathématiques. Cela signifie que la succession des préalables que requiert l'habileté terminale doit être établie. Les autres éléments d'apprentissage (par exemple les informations verbales, les habiletés motrices, les attitudes) peuvent être présentés dans n'importe quel ordre, pourvu que l'on tienne compte de la nature particulière de la situation d'apprentissage.

Puisque notre intervention se situe dans le domaine des sciences, nous avons opté pour une position mitoyenne. A savoir, nous utilisons une hiérarchie de préalables d'habiletés intellectuelles ayant comme but de faire satisfaire des objectifs terminaux. Cependant, nous ne fixons pas une fois pour toutes la séquence d'apprentissage. Nous la brisons en des situations d'apprentissage qui ont des tailles les plus modestes possibles et qui délivrent des objectifs d'apprentissage les plus significatifs possibles par rapport aux objectifs terminaux. Pour poursuivre la métaphore précédente de l'atome, on pourrait dire que nous organisons la matière à apprendre en molécules et qu'ensuite nous les articulons aux moyens de structures.

3. ARCHITECTURE DU SYSTEME

Le système informatique de conception et de réalisation de logiciels éducatifs que nous proposons est composé d'éléments, d'une structure sur ces éléments, ainsi que d'une articulation de cette structure.

Le système informatique est composé des modules suivants:
- le module des connaissances;
- le module des métaconnaissances;
- le module dossier-apprenant;
- le module tuteur;
- le module évaluateur;
- le module spécificateur.

Pour soutenir l'analyse du système, donnons une représentation graphique de celui-ci.

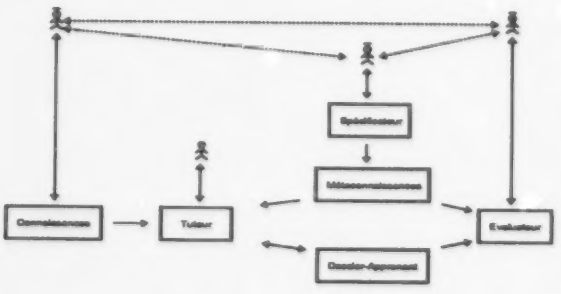


Fig. 1 Schéma du système

Les lignes en traits pleins indiquent le sens du flot d'informations circulant entre les humains, et/ou entre les différents modules du système.

Le système informatique modélisé par la figure 1 a quatre interfaces. Trois de ces interfaces concernent soit le réalisateur du logiciel

éducatif, soit son gestionnaire. Ces interfaces sont dans les modules évaluateur et spécificateur ainsi que dans le module des connaissances. L'interface qui concerne l'apprenant est dans le module tuteur.

Dans la présentation du système, une interface n'est pas développée. C'est celle joignant le module des connaissances au réalisateur ou au gestionnaire. Cette interface n'est pas développée puisque nous supposons que cette personne utilisera un système-auteur, un langage d'auteur ou tout autre langage informatique capable de construire et de modifier des situations ou molécules du module des connaissances.

Reprenons chacun des modules du système pour établir leurs objectifs propres et pour les mettre en relation avec la théorie de l'apprentissage de R. M. Gagné.

3.1 Le module des connaissances

Le module des connaissances, comme son nom l'exprime, regroupe l'ensemble des connaissances qu'a le logiciel éducatif sur le sujet traité.

Le module des connaissances est organisé par rapport aux phases du processus d'apprentissage ainsi que par rapport à la nature de la matière à apprendre.

Trois types d'activités sont prévus pour tenir compte des phases du processus d'apprentissage. Ce sont:
- les activités facilitant l'appropriation de la matière,
- celles assurant la consolidation des apprentissages et
- celles assurant le transfert nécessaire des apprentissages pour résoudre effectivement des problèmes.

Les activités qui facilitent l'appropriation de la matière se situent généralement à l'amorce d'un nouveau cycle d'apprentissage d'un thème de la matière ou lors d'un nouveau cycle de l'apprentissage d'un thème déjà vu. Elles se situent normalement au début du processus d'apprentissage et visent les produits de l'apprentissage reliés aux objectifs cognitifs de la partie inférieure de leur taxonomie. Les produits d'apprentissage concernés sont la discrimination, les concepts et les règles.

Les activités qui assurent la consolidation des apprentissages visent les mêmes produits de l'apprentissage que les activités précédentes, sauf qu'elles se situent à la partie médiane ou finale d'un cycle du processus d'apprentissage.

Les activités qui assurent le transfert nécessaire des apprentissages pour résoudre effectivement des problèmes se situent d'emblée à la fin d'un cycle du processus d'apprentissage et visent les produits de l'apprentissage reliés aux objectifs cognitifs de la partie supérieure de leur taxonomie. Ces produits sont les règles, les règles d'ordre supérieur ainsi que les stratégies cognitives.

Comme les domaines d'application du système informatique contiennent des habiletés intellectuelles ainsi que le développement de stratégies cognitives, la matière est divisée en un ensemble de situations d'apprentissage de taille minimale avant des objectifs spécifiques. Ce sont les molécules de connaissance du système. La dimension des situations est maintenue la plus petite possible pour permettre le plus de flexibilité au niveau de leurs articulations. Une situation peut être, pour fixer les idées, une séquence et/ou une combinaison de pages-écrans interactives concernant l'apprentissage d'un concept à l'aide de concepts sous-jacents. Chaque situation peut délivrer des objectifs d'apprentissage. De même, chaque situation possède une structure d'habiletés intellectuelles prérequises que l'apprenant doit satisfaire. Schématiquement, on peut représenter une situation d'apprentissage comme suit:

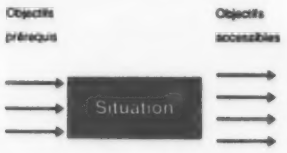


Fig. 2 Schéma d'une situation d'apprentissage

3.2 Le module des métaconnaissances

Le module des métaconnaissances sert à structurer les connaissances. Il les structure par rapport aux objectifs terminaux d'apprentissage que l'on veut faire atteindre à l'apprenant ainsi que par rapport à certains objectifs spécifiques d'apprentissage. Il est composé des éléments suivants:
- une hiérarchie des préalables d'habiletés intellectuelles;
- un plan d'étude et
- un ensemble de structures de prérequis des situations.

Le module des métaconnaissances structure les connaissances par rapport aux objectifs terminaux au moyen de la hiérarchie des préalables d'habiletés intellectuelles. Les objectifs d'apprentissage servent ainsi à articuler l'ensemble des situations d'apprentissage. A titre d'exemple, cette structure pourrait être illustrée par la hiérarchie de la figure 3.

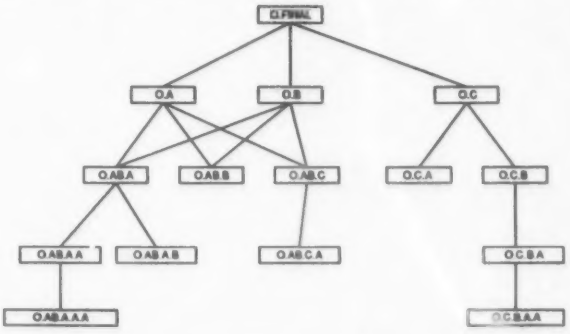


Fig. 3 Exemple d'hiérarchies d'objectifs

Le module des métaconnaissances structure aussi les connaissances par rapport à certains objectifs spécifiques que l'on veut faire atteindre à l'apprenant. Ceci est possible grâce d'une part, au plan d'étude et, d'autre part, à l'ensemble des structures de prérequis des situations.

Le plan d'étude permet de donner au logiciel éducatif une métaconnaissance synthétisante de la matière, en établissant une articulation spéciale entre les diverses situations d'un sous-ensemble de l'ensemble des situations. Par exemple, si l'ensemble des situations est $\{S_1, S_2, \ldots S_{11}\}$, alors une structuration peut être:
$((S_6$ ET $S_8)$SQ$(S_9$ OU $S_{10}))$
OU $((S_1$ ET S_2 ET $S_3)$SQ$(S_4$ OU $S_5))$.
où SQ signifie que l'on poursuit en séquence (cf. Figure 4).

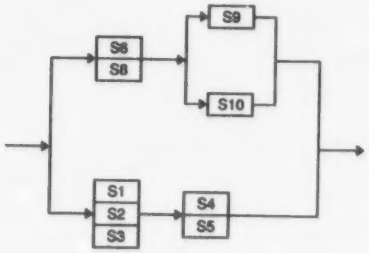


Fig. 4 Exemple de plan d'étude

Un plan d'étude peut être soit vide, soit incomplet, soit complet. On dit d'un plan d'étude qu'il est complet lorsque sa réalisation conduit à la satisfaction de tous les objectifs terminaux. Autrement, il est dit incomplet. Évidemment, un plan d'étude vide est incomplet.

L'ensemble des structures de prérequis des situations concourt aussi à articuler l'ensemble des situations, puisque certains objectifs délivrés par une situation peuvent servir de prérequis pour une autre situation. Les prérequis de situations d'apprentissage sont donc de même nature que les objectifs délivrables par une situation d'apprentissage.

3.3 Le module dossier-apprenant

Le dossier-apprenant est la base de données qui recueille les informations concernant les interactions de l'apprenant avec le logiciel éducatif. Il permet l'individualisation de l'approche du tuteur.

3.4 Le module tuteur

Le module tuteur est la portion du logiciel éducatif qui s'occupe des interactions avec l'apprenant. Il a comme objectifs fondamentaux de fournir à l'apprenant
- des interactions de qualité,
- une rétroaction en temps réel concernant ses apprentissages,
- de l'information individualisée en temps réel concernant les activités d'apprentissage,
- des métaconnaissances sur les façons d'opérer du logiciel face à l'apprentissage de la matière à l'étude.

Comme le rappelle la problématique, une des carences des didacticiels classiques est de fournir une articulation des situations d'apprentissage qui se situe presque uniquement au niveau global. Dans un tel contexte, il est très difficile, sinon impossible, d'articuler finement les situations d'apprentisssage. Avec l'approche proposée, il est possible de réaliser cette articulation
- à l'aide de la page de gestion de l'apprentissage et
- par l'introduction de modes différenciés de fonctionnement.

L'objectif fondamental de la page de gestion de l'apprentissage est de fournir à l'apprenant la possibilité de prendre en main son apprentissage. Pour ce faire, la page de gestion de l'apprentissage accorde la possibilité à l'apprenant de connaître le cheminement qu'il a effectué dans l'ensemble des situations, les résultats qu'il a obtenus, le plan d'étude fournit par l'enseignant, la structure des objectifs d'apprentissage et les statistiques associées à chaque objectif. Il a aussi accès à la structure de prérequis de chaque situation ainsi qu'à la liste des objectifs de chaque situation.

La page de gestion de l'apprentissage est aussi le lieu où l'apprenant peut choisir le mode d'interaction qu'il désire avoir avec le logiciel éducatif. C'est à cet endroit que l'apprenant peut choisir de fonctionner soit
- en mode exploratoire,
- en mode autonome ou,
- en mode dirigé.

En mode exploratoire, l'apprenant peut visiter toutes les situations d'apprentissage du logiciel éducatif. Cette option est offerte pour permettre à l'utilisateur de se familiariser avec le contenu du logiciel et pour lui permettre de mieux comprendre sa structure. Les prérequis et les résultats des situations sont ignorés. Cependant, le logiciel se souvient du nombre de fois que l'apprenant a parcouru toute situation dans ce mode. Ces informations servent comme indicateur pour déterminer le style d'apprentissage qu'adopte l'élève lors de l'utilisation du logiciel.

Pour entrer dans une situation en mode autonome, l'utilisateur doit avoir atteint ou dépassé le seuil de satisfaction de la structure des prérequis de cette situation. L'apprenant est informé de la ou des raisons qui l'empêchent d'entrer dans une situation. Les résultats générés par la situation sont gardés de la même façon qu'en mode dirigé. En fait, le tuteur fonctionne de la même façon qu'en mode dirigé sauf qu'il suggère des alternatives, et, qu'en définitive, c'est l'apprenant qui fait le choix.

En mode dirigé, le tuteur vise comme toujours la satisfaction par l'élève des objectifs terminaux d'apprentissage. Pour atteindre ce but, il essaie de lui faire satisfaire étape par étape des sous-ensembles d'objectifs. Les objectifs de chaque étape sont déterminés, en respectant l'ensemble des structures des prérequis des

situations, soit par le plan d'étude, soit par la structure hiérarchique des objectifs d'apprentissage.

Une autre des carences des didacticiels classiques consiste à ne pas fournir une rétroaction suffisante sur les apprentissages déjà faits. Tout au plus, dans le meilleur des cas, se contente-t-on de fournir en différé la possibilité de revenir sur les apprentissages. Le tuteur permet une authentique rétroaction en donnant à l'apprenant l'accès
- à son cheminement dans l'ensemble des situations d'apprentissage et
- à ses résultats structurés par rapport à la hiérarchie des préalables.

Le tuteur, par l'intermédiaire de la page de gestion de l'apprentissage, donne l'accès à l'apprenant au détail du cheminement qu'il a parcouru dans le logiciel éducatif. Cette information est tant qualitative que quantitative.

Qualitativement, le cheminement donne l'ordre et le mode avec lequel l'apprenant a parcouru chaque situation. Quantitativement, le cheminement fournit la contribution effective de chaque situation à l'obtention des objectifs terminaux dans le contexte où elle a été sélectionnée.

Le tuteur donne aussi à l'apprenant l'accès à ses résultats structurés par rapport à la hiérarchie des objectifs d'apprentissage. Ainsi, l'utilisateur peut suivre en temps réel l'évolution de ses apprentissages.

A la fin de l'exécution de chaque situation d'apprentissage, le tuteur évalue l'ensemble des situations et choisit au moyen de stratégies, celle convenant le mieux à l'apprenant. Ces stratégies s'appuient sur
- le plan d'étude de l'ensemble des situations,
- la structure des objectifs d'apprentissage,
- l'ensemble des structures des prérequis des situations,
- les résultats de l'apprenant ainsi que sur
- le cheminement spécifique de celui-ci.

Le tuteur fournit aussi à l'apprenant des métaconnaissances sur les façons d'opérer du logiciel éducatif face à l'apprentissage de la matière à l'étude
- du point de vue des objectifs d'apprentissage ainsi que
- du point de vue des situations d'apprentissage.

Du point de vue des objectifs d'apprentissage, le tuteur fournit à l'élève la structure hiérarchique et la liste globale des objectifs d'apprentissage.

Du point de vue des situations d'apprentissage, le tuteur donne à l'élève l'accès à la liste globale des situations d'apprentissage, à la liste des objectifs délivrables par une situation, à la structure des prérequis de chaque situation et enfin à la liste des situations délivrant un objectif déterminé.

3.5 Le module évaluateur

L'évaluateur est la partie du logiciel éducatif qui permet à son gestionnaire de questionner la base de données dossier-apprenant. La présentation des informations est structurée à partir des renseignements contenus dans le module des métaconnaissances. L'objectif de l'évaluateur est de permettre au gestionnaire de poser, dans le contexte d'activités d'enseignement-apprentissage réalisés à l'aide du logiciel éducatif, des diagnostics concernant les apprentissages. Il peut donc servir à faire des évaluations tant formatives que sommatives.

3.6 Le module spécificateur

Le spécificateur est la section du logiciel éducatif qui permet à son réalisateur ou à son gestionnaire d'entrer et/ou de modifier les métaconnaissances concernant la matière à l'étude. Le spécificateur donne ainsi une plus grande souplesse d'utilisation au logiciel éducatif en lui permettant de s'adapter aux situations réelles d'enseignement-apprentissage.

3.7 Fonctionnement du système

Le système peut être fractionné en deux parties. Celle orientée vers l'apprenant et celle orientée vers le réalisateur/gestionnaire du logiciel éducatif.

La partie du système orientée vers l'apprenant est composée de connaissances sur la matière, de connaissances sur l'apprenant, de métaconnaissances sur la matière et d'un tuteur. Le moteur du système est le tuteur. L'objectif du tuteur consiste à faire satisfaire à l'apprenant tous les objectifs terminaux d'apprentissage. Pour ce faire, il utilise des structures, des stratégies et des activités d'apprentissage.

La partie du système orientée vers le réalisateur/gestionnaire du logiciel éducatif est composée de l'évaluateur et du spécificateur. Le spécificateur sert à définir les métaconnaissances sur la matière. L'évaluateur permet au gestionnaire de vérifier l'adéquation de fonctionnement du système à l'aide des connaissances sur l'apprenant, des métaconnaissances sur la matière et des stratégies du tuteur. Si cette adéquation venait à faire défaut, le gestionnaire pourrait utiliser soit le spécificateur, soit un système-auteur pour corriger la situation.

4. CONCLUSION

Citons d'abord quelques caractéristiques de l'approche proposée. Non seulement la réalisation du système informatique d'assistance à la conception et à la réalisation au niveau des métaconcepts sous-jacents aux logiciels éducatifs utilise-t-elle des techniques d'intelligence artificielle, mais ces mêmes techniques sont transférées dans les logiciels éducatifs produits par ce système de façon à y être opérationnelles.

Cette caractéristique a pour effet de rendre beaucoup plus versatile l'utilisation d'un logiciel éducatif construit avec ce système. En effet, pour se conformer à une modification du curriculum ou pour s'ajuster à une modification du plan de cours, il suffit de modifier soit le module des connaissances, soit le module des métaconnaissances, soit les deux pour obtenir un nouveau logiciel éducatif. Le système autorise donc l'expérimentation.

Une autre caractéristique digne de mention des logiciels éducatifs produits à l'aide du système proposé consiste à exhiber la démarche didactique ainsi que les approches psychopédagogiques des auteurs du logiciel éducatif. Cette caractéristique est fondamentale lorsque l'utilisateur du logiciel éducatif est un enseignant qui désire parfaire ses connaissances sur le sujet traité par le logiciel éducatif.

Aux aspects novateurs précédents s'ajoutent des améliorations fonctionnelles pour les logiciels éducatifs produits par le système proposé. L'assouplissement des articulations de l'architecture de ces logiciels éducatifs est tel que la rétroaction en temps réel de l'apprenant sur ses apprentissages est possible dans le cadre même du logiciel éducatif.

RÉFÉRENCES

1. Lavoie M., Gagné M., Jacques A., Dufour D., Beaulieu V. Logiciel éducatif ouvert et complet concernant l'apprentissage de la logique séquentielle (86-RPS-01). Programme RDF, DGERU, Ministère de l'enseignement supérieur et de la science, Québec, Canada, 1986.

2. Carton J.L., Quéré M. Des outils pour l'E.I.A.O., le projet MEDIAN. Rapport FINAL DE LA CONVENTION ADI 83/372, France, 1987.

3. Fine S.E. APT Reference manual, Hazeltine Corp., Reston, VA, U.S.A., 1985.

4. Mudrick D., Stone D. An adaptive Authoring System for Computer Based Instruction, J. Computer-Based Instruction, vol. 11, nº 3, 1984, p. 82-84.

5. Regourd J.P. GAMETE: le manuel de référence. Rapport LITP nº 87.9, Université P. et M. Curie, Paris, France, 1987.

6. Wenger E. Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge. Morgan Kaufman Pub. Co., Los Altos, CA, U.S.A., 1987.

7. Lavoie M., Beaulieu V., Un modèle didactique servant à la conception, à l'élaboration et à la réalisation de logiciels éducatifs. L'enseignement assisté par ordinateur. Les cinq prochaines années. Compte rendu du 5e symposium canadien sur la technologie pédagogique, Conseil National de Recherches Canada, Ottawa, Canada, 1986, p. 375-378.

8. Barril P., De La Passardière B., Le Calvez F., Madaule F., Poc M., Urtassun M. Quels outils informatiques pour réaliser des didacticiels? Étude comparative. Rapport MASI (Méthodologie et Architecture des Systèmes Informatiques), nº 93, Institut de Programmation, Université P. et M. Curie, Paris, France, 1985.

9. De La Passadière B. Quels outils informatiques pour réaliser des didacticiels? Une étude sur ARLEQUIN-DIANE. Rapport MASI, nº 94, Institut de Programmation, Université P. et M. Curie, Paris, France, 1985.

10. Poc M. Quels outils informatiques pour réaliser des didacticiels? Une étude sur SUPERPILOT. Rapport MASI, nº 109, Institut de Programmation, Université P. et M. Curie, Paris, France, 1985.

11. Le Calvez F. Quels outils informatiques pour réaliser des didacticiles? Une étude sur DUO. Laboratoire MASI, Rapport nº 116, Institut de Programmation, Université P. et M. Curie, Paris, France, 1986.

12. Brouard J. Évaluation d'une application de l'enseignement assisté par ordinateur en milieu de travail. Conseil National de Recherches Canada, ERB-982, Rapport nº 25254, Ottawa, Canada, 1985.

13. Gagné R.M., Briggs L.J. Principles of Instructional Design, Holt, Rinehart and Winston, New York, U.S.A., 1979.

14. Romiszowski A.J. Designing Instructional Systems. Decision making in course planning and curriculum design. Kogan Page, London, G.B., 1981, p. 176-177.

15. Beeson M.G. Hierarchical learning in electrical science, Journal of research in Science Teaching, vol. 14, nº 2, 1977, p. 117-127.

THERMEX: SYSTÈME-EXPERT APPLIQUÉ À LA RÉSOLUTION DE PROBLÈMES EN THERMODYNAMIQUE

L. Thérien
Faculté d'éducation
Université de Sherbrooke
Sherbrooke, PQ

B. Marcos, A. Boily, J. Lapointe, M. Veillette
Faculté des sciences appliquées
Université de Sherbrooke
Sherbrooke, PQ

Cet article présente le dévelopement et les premières expérimentations d'un système-expert devant assister les étudiant(e)s dans la résolution de problèmes en thermodynamique.

Cet article va d'abord décrire comment les connaissances à propos de la matière (thermodynamique) peuvent se représenter sous forme de règles et d'objets. Nous avons observé les étudiant(e)s (magnétoscope) en situation de résolution de problèmes et nous avons déduit comment modéliser leur méthodes de résolution de problèmes à l'aide d'une suite de formules et de résultats intermédiaires. Cet article indiquera comment le système-expert intervient lorsqu'il a diagnostiqué une démarche erronée de l'étudiant(e). Le diagnostic est bâti à partir de l'inventaire des principales erreurs commises dans ce type de problèmes.

Le système-expert obtenu est écrit en Prolog et fonctionne sur IBM-PC. Les premières expérimentations réalisées lors du cours de thermodynamique seront discutées.

THERMEX: AN EXPERT SYSTEM APPLIED TO THE SOLUTION OF PROBLEMS IN THERMODYNAMICS

This article will describe the development and first experiments with an expert system for assisting students in the solution of thermodynamics problems.

It will first relate how knowledge on the subject (thermodynamics) can be represented in the form of rules and objects. We have observed students (in video recordings) in problem-solving situations and have deduced how to model their methods of solving problems using a series of formulas and intermediate results. This article will indicate how the expert system intervenes when it has diagnosed a faulty approach by the student. The diagnosis is developed on the basis of the list of main errors made in this type of problem.

The expert system prepared is written in Prolog and operates on an IBM-PC. The first experiments performed during the thermodynamics course will be discussed.

THERMEX: SYSTÈME-EXPERT APPLIQUÉ À LA RÉSOLUTION DE PROBLÈMES EN THERMODYNAMIQUE

L. Thériene, B. Marcos*, A. Boily*
J. Lapointe* et M. Veillette*

*Faculté d'éducation,
*Faculté des sciences appliquées
Université de Sherbrooke
Sherbrooke, Qué., J1K 2R1

INTRODUCTION

Parmi des cours dispensés dans une faculté de génie, la thermodynamique constitue une matière difficile pour beaucoup d'étudiant(e)s. La Faculté des sciences appliquées de l'Université de Sherbrooke ne fait pas exception à cette règle. Pour remédier à cette situation, plusieurs solutions ont été envisagées parmi lesquelles un renforcement du rôle des assistants. Après avoir questionné les étudiants, il est apparu que le support fourni par les assistants n'était pas toujours adéquat pour les aider à résoudre les problèmes de thermodynamique. Nous avons donc proposé un système-expert pour améliorer le support fourni aux étudiants. Cet article présente les diverses composantes d'un système-expert qui a pour but d'assister les étudiants dans la résolution de problèmes thermodynamiques.

Depuis plusieurs années, de nombreux systèmes d'Enseignement Intelligemment Assisté par Ordinateur (EIAO) ont été développés. En se limitant à des matières enseignées au niveau universitaire dans le domaine du génie, on peut citer les travaux de Slater (1987)[1] pour l'enseignement de la statique, de Bottino (1986)[2] pour l'enseignement de la théorie des groupes et de Paries (1988)[3] pour l'apprentissage des notions de base en électricité. Dans le domaine de l'enseignement de la thermodynamique, des systèmes d'EAO ont déjà été proposés: Vermette (1986)[5], à partir d'une banque de problèmes résolus, suit l'étudiant(e) dans sa démarche de résolution; cependant son logiciel a peu de facilité d'explications et est assez rigide dans l'écriture des diverses relations.

Nous désirons, avec la construction de THERMEX, offrir un environnement d'aide à la résolution de problèmes qui puisse fournir une aide adaptée à l'étudiant(e) qui est en train de résoudre un problème de thermodynamique. THERMEX doit aussi fournir des explications sur les erreurs des étudiant(e)s et proposer une action corrective appropriée.

Avant de décrire les composantes de THERMEX, il est important de rappeler les remarques suivantes. THERMEX est utilisé pour aider à la résolution de problèmes et non pour enseigner la matière. Il faut donc privilégier la composante diagnostic. THERMEX est utilisé principalement par les étudiant(e)s ayant quelques difficultés pour résoudre les exercices. On suppose enfin que l'étudiant(e) a tenté de résoudre l'exercice et a quelques éléments de solution.

ARCHITECTURE DE THERMEX

Comme la plupart des systèmes d'EIAO, THERMEX comprend les éléments suivants:

- un module contenant l'expertise à propos du domaine,
- un module contenant les stratégies tutoriales,
- un module contenant le modèle de l'étudiant.

EXPERTISE À PROPOS DU DOMAINE

La partie de la thermodynamique enseignée dans les facultés de génie peut se représenter sous une forme hybride regroupant des objets et des règles de production. En suivant la terminologie classique de la représentation-objet, on peut définir les classes suivantes. La classe principale est le système thermodynamique divisé en plusieurs sous-classes. La sous-classe ETAT a pour objet les états thermodynamiques caractérisant le processus. Les attributs de ces objets sont les variables d'état comme la pression, la température, le volume, l'enthalpie, l'énergie interne, etc... La sous-classe TRANSFORMATION a pour objet les différentes transformations opérant sur le système; ces objets ont les attributs précisant la nature de la transformation. Ainsi la transformation peut être isobare, isochore ou isotherme et se produire entre l'entrée et la sortie.

Une autre sous-classe CONDITION contient les conditions d'opération de chacune des procédures définies précédemment. Par exemple, la relation donnant le travail au cours d'une transformation adiabatique et réversible va être reliée à l'objet CONDITION d'opération de cette relation qui va avoir comme attributs adiabatique et réversible.

Les classes possèdent aussi des procédures qui vont permettre de traiter les différents attributs. Ces procédures traduisent en particulier les théorèmes de la thermodynamique comme les bilans de masse et les bilans d'énergie, les expressions du travail ou de la chaleur, le second principe de la thermodynamique. Ces procédures

peuvent s'appliquer à des attributs de différents objets comme celles citées précédemment ou des attributs du même objet comme la loi des gaz parfaits. Ces procédures sont représentées sous forme de règles de production.

La base de connaissances relative à la thermodynamique comprend aussi des règles précisant la stratégie à suivre pour définir complètement le système thermodynamique. Un exemple d'une de ces règles est: si on connaît (n-1) variables d'une relation ayant n variables, on peut calculer la dernière variable; ou encore, si trois attributs d'un objet ETAT sont connus, tous les autres attributs sont calculables. Il est donc possible, après avoir rentré les différentes hypothèses du système physique, de déduire toutes les conséquences de ces hypothèses en utilisant par exemple un moteur d'inférences du type PROLOG. Cependant, ces déductions obtenues de manière formelle ne sont pas très utiles car les raisonnements pour y arriver ne sont pas en général ceux effectués par l'expert (le professeur). Ils sont en général plus lourds que ceux suivis par le professeur et d'un intérêt limité. De plus, ils ne sont pas directement utilisables dans le processus de diagnostic, ce qui constitue un des buts principaux de THERMEX.

La démarche suivie par le professeur et qui est par exemple disponible dans le corrigé est câblée directement en utilisant les formes de représentation de la base de connaissances. Cette démarche sera utilisée dans le module tuteur.

MODÈLE DE L'ÉTUDIANT

Dès le développement des premiers systèmes d'EIAO, la composante modèle de l'étudiant(e) est apparue comme un élément très important de ce type de logiciel. Parmi les modèles classiques, on peut citer "l'overlay Model"[6] qui modélise l'étudiant(e) comme un sous-ensemble du modèle de l'expert, le "Bug Model"[7] qui identifie l'étudiant(e) à des modèles d'erreurs pré-câblés ou les modèles dynamiques qui construisent au cours de la session une représentation de l'étudiant(e)[3,8].

Avant de choisir le modèle, nous avons observé pendant une session un groupe d'étudiant(e)s en situation de résolution de problèmes. Nous avons ainsi relevé les principales erreurs commises que nous avons ensuite classées en plusieurs catégories: erreurs dans la représentation du problème, mauvaises connaissances des concepts, erreurs dans le choix et l'application des relations (mauvaises conditions d'application), erreurs d'unités et erreurs de calcul. De ces observations, nous proposons que le modèle de l'étudiant soit formé des éléments suivants: les hypothèses faites (ou oubliées), les résultats numériques obtenus et les relations utilisées (ou non). Le modèle étudiant est relatif à un problème donné. Le modèle est représenté avec le même formalisme que celui utilisé dans la base de connaissances décrivant le domaine de la thermodynamique. Ce modèle n'appartient pas à la famille des "Overlay Model" car il ne se définit pas comme un sous-ensemble de la démarche de l'expert. Cependant ce modèle peut être comparé à la représentation de la démarche de l'expert, car le même formalisme de représentation est utilisé. Il n'appartient pas non plus à la famille des "bug model", car tous les modèles d'erreurs découverts dans la phase d'observation, n'ont pas été préalablement rentrés. Cependant certains modèles d'erreurs significatifs ont été introduits pour accélérer la procédure de diagnostic. S'il a des caractéristiques des deux familles précédentes, le modèle étudiant retenu se construit dynamiquement au cours de la session d'assistance.

DÉMARCHE TUTORIALE

Le module tuteur gère la présentation des différents exercices, pose un diagnostic et propose une démarche corrective.

Le modèle étudiant sert de point de départ à la procédure de diagnostic. La première étape analyse les réponses de l'étudiant(e) et les compare avec les réponses exactes. La deuxième étape vérifie la consistance du raisonnement c'est-à-dire si les hypothèses (justifiées ou erronées) et les concepts employés (bien ou mal) conduisent aux résultats obtenus par l'étudiant(e). Dans cette deuxième étape, nous utilisons comme mécanisme d'inférences les mécanismes exacts utilisés par les enseignants ou quelques mécanismes erronés fréquemment observés. Dans ce cas, ces mécanismes erronés correspondent à un élément d'un bug model. Si les résultats sont inconsistants et s'il n'y a aucun modèle erroné adéquat, la deuxième étape se révèle inefficace. Par contre la première étape fournit toujours de l'information.

A partir des informations obtenues dans la démarche de diagnostic, le module tuteur va simuler la démarche d'un assistant chargé de procurer de l'aide. Le modèle tuteur détermine jusqu'où la résolution a été poussée. Si un élément dans cette démarche s'avère incorrect, le module tuteur réagit suivant la nature de cet élément. Si un concept important est manquant pour résoudre l'exercice, le module tuteur signale qu'il y a un oubi et aide l'étudiant(e) à l'identifier pour qu'il puisse continuer. Comme aide, le tuteur rappelle la définition des concepts de la famille du concept manquant, donne des exemples et demande à l'étudiant d'identifier le bon concept. Les concepts importants sont caractérisés à partir de l'expérience des enseignants. S'il n'y a pas de concepts importants absents ou erronés, le tuteur laisse exposer l'étudiant en notant les éléments erronés pour les lui révéler à la fin de la session; ces éléments sont par exemple des hypothèses superflues, l'emploi de relations inutiles, une mauvaise définition d'un système, des erreurs d'unités ou de calcul. L'analyse de la consistance des résultats permet de révéler des mécanismes erronés. Si on les identifie avec un modèle d'erreurs, le modèle tuteur informe l'étudiant de cette situation et explique pourquoi ce mécanisme est erroné. Par exemple, si l'étudiant(e) mélange les états d'entrée-sortie d'un système avec les états de début et fin d'une transformation, les aspects d'un système ouvert en régime permanent seront détaillés. Si le mécanisme erroné n'est

pas identifié, le module tuteur va vérifier les hypothèses et les concepts présents et absents et à partir de ces éléments pousser l'étudiant vers des éléments de la solution. Dans ce cas, le module tuteur utilise d'abord les réponses correctes fournies par l'étudiant(e). S'il n'en existe aucune, le tuteur lui propose des éléments de la démarche de résolution utilisée par les professeurs.

UTILISATION

THERMEX a été essayé de manière expérimentale au cours de la dernière session. L'expérimentation rapportée ici ne concerne que la réponse des étudiant(e)s au système-expert. On n'a donc pas mesuré l'influence du logiciel sur les performances des étudiant(e)s.

L'observation des étudiant(e)s a montré que ces derniers emploient des concepts sans les préciser explicitement. Comme THERMEX les oblige à le faire, ils ont donc pu identifier les oublis ou les contre-sens qu'ils commettent sur l'interprétation des concepts. Nous avons aussi relevé que des relations sont souvent utilisées sans en préciser les conditions d'opération. Cet oubli conduit à l'emploi de relations inappropriées au contexte. THERMEX les restreint alors dans le choix des relations possibles en les obligeant à respecter les conditions d'opération de ces relations. Si des conditions opératoires proposées sont erronées cette erreur indique une mauvaise compréhension d'un concept ou d'une relation. Si des conditions opératoires sont oubliées, cette omisson indique une mauvaise lecture du texte ou une mauvaise représentation. THERMEX signale certaines erreurs numériques ce qui est révélateur pour l'étudiant(e): ce sont, par exemple, les erreurs de signe sur des quantités comme le travail ou la chaleur, montrant que l'étudiant(e) n'a pas questionné la signification physique des résultats obtenus.

Ces observations ont montré que THERMEX fournit des informations non disponibles sur un manuel d'exercices avec corrections[9]. Les observations ont montré enfin que les remarques et corrections dépendent du cheminement de chaque étudiant.

DISCUSSION ET RECOMMANDATION

Le module tuteur de THERMEX impose aux utilisateurs une approche pour la résolution des différents exercices. Ainsi les questions et les réponses se font à travers des menus. Cette approche par menu nous permet d'éviter la difficulté d'interprétation d'une réponse en langage naturel. A cause de la grande variété de combinaison possible de réponses, il est peu probable que l'étudiant pointe des réponses complètement au hasard. Les menus évitent aussi l'analyse de l'écriture des relations. Nous éliminons ainsi les erreurs de transcription que commettent les étudiant(e)s lorsqu'ils recopient une relation. Cependant, ces erreurs d'inattention ne sont pas assez importantes qualitativement et quantitativement pour devoir être incorporées dans le système d'EIAO. Les modèles d'étudiants "erratiques" n'ont pas non plus été incorporés même s'ils ont été observés à l'occasion; un exemple d'un tel comportement est celui d'un étudiant(e) inventant des valeurs numériques pour certaines propriétés d'état lorsqu'il est bloqué. Même si ces comportements erratiques correspondent en général à des représentations mentales de l'exercice, il a été difficile de préciser ces représentations et donc de les modéliser.

Le système d'EIAO THERMEX est encore au stade préliminaire et doit être amélioré. La banque de modèles étudiants erronés (bug model) est assez limitée. Les modèles qu'elle contient sont en général formés de quelques règles fausses qui sont des variantes des procédures utilisées par les enseignants. Cette banque devra être enrichie par l'ajout de nouveaux modèles et de modèles plus riches. Ces améliorations seront faites à l'aide des expériementations futures.

La mise au point de THERMEX va être poursuivie par une étape de validation qui sera faite à la prochaine session ou sera dispensée dans le cours de thermodynamique. La validation sera faite en choisissant un groupe témoin pour pouvoir mesurer l'influence du système d'EIAO sur les performances des étudiant(e)s. Enfin l'approche proposée dans cet article pourra facilement être adaptée à d'autres matières que la thermodynamique si ces matières peuvent se représenter à l'aide du formalisme hybride (objet + règles de production).

REMERCIEMENT

Nous tenons à remercier la compagnie IBM CANADA et la Faculté de Génie de l'Université de Sherbrooke, pour le support financier qu'elles ont accordé à ce projet. Nous tenons aussi à remercier les étudiant(e)s qui ont accepté de tester ce logiciel.

RÉFÉRENCES

1. Slater, J.H., M. Ahuja, Building an Intelligent Tutor Engineering: the Macavity Experience. Knowledge Based Expert System for Engineering: Classification, Education and Control, D. Sriram and R. Adey (Editors), pp 157-173, August 1987.

2. Bottino, R.K., Forcheri, P. and Molfino, M.T., Developping Educational Software: An approach of AI perspective. Cinquième symposium canadien sur la technologie éducative, pp 78-83, 1986.

3. Paries, O., Diagnostic du comportement cognitif et apprentissage symbolique. Proceedings of Intelligent Tutoring System, Montreal, pp 201-205, Juin 1988.

4. Vermette, J., Camarero, R., Thermodynamique Assistée par Ordinateur. Proceedings of the Fifth Canadian Conference on Engineering Education, London, pp 124-139, May 1986.

5. Klein, S.A., CP/THERMO F-Chart Software, Instructor's Manual. University of Wisconsin, Madison, 1985.

6. Carr, B. and Goldstein, I., Overlays: a theory of modeling for computer aided instruction. Artificial Intelligence Meno 406, MIT 1977.

7. Brown, J.S. and Burton, R.R., Diagnosis Models for procedural bugs in basic mathematical skills, Cognitive Sci. 2, pp 155-192, 1978.

8. Ikeda, M., Mizoguchi, R. and Kakusho, O., Design of a general frameworks of ITS. Proceedings of Intelligent Tutoring System, Montreal, pp 82-89, Juin 1988.

9. Becke, L.A., Computer-Aided Misconception Based Intelligent Tutoring and Exercise Generation. Journal of the Association for Educational and Training Technologies, 25(1), 68-73, 1988.

KNOWLEDGE-BASED OR EXPERT SYSTEMS - A NEW TOOL FOR ENVIRONMENTAL IMPACT ASSESSMENT INSTRUCTION

A. Gray, C. Hickman
School for Resource and Environmental Studies
Dalhousie University
Halifax, NS

Once confined to large industrial and academic research institutions due to the large commitment of high-cost computing resources required, knowledge-based or expert systems (KBS/ES) technology has now become available to most micro-computer users. Such systems allow the capture of expert knowledge concerning a topic, which may then be given a wider distribution.

Another important function of KBS/ES is that of training or instruction. The format of many such systems, particularly those developed using software that is termed an "expert system shell", offers guidance to the user, and has the ability, by examining the logic rules used by the system, to follow the detailed steps required to reach a conclusion. The values of key variables may be changed, and another run may then be made to see how the outcome is altered. The nature of KBS/ES offers particular benefits in the field of environmental impact assessment, a multi-disciplinary process that encompasses bio-physical, socio-economic and political considerations. Knowledge-based or expert systems can deal with qualitative reasoning and reasoning based on uncertain knowledge - an important consideration in many assessments. The system can also be linked to external programs, such as models, that can deal with aspects of the situation that involve quantitative understanding.

The key concepts of such systems as outlined above, are illustrated with reference to a system developed to determine the impact of culvert installations on fish habitat. A planned use of the system is as a self-instructional tool in a graduate course in environmental impact assessment.

SYSTÈMES EXPERTS - NOUVEL OUTIL DE FORMATION DANS LE DOMAINE DES EFFETS SUR L'ENVIRONNEMENT

Autrefois réservée aux importants centres de recherche industriels et académiques en raison du coût élevé des ressources informatiques qu'elle impliquait, la technologie des systèmes experts est désormais à la portée de la plupart des utilisateurs de micro-ordinateurs. Ces systèmes permettent de saisir les connaissances d'experts d'un domaine, connaissances qui peuvent alors connaître une plus grande diffusion.

Les systèmes experts jouent également un rôle important dans la formation. La structure d'un bon nombre de ces systèmes, principalement ceux qui ont été élaborés au moyen d'un logiciel qu'on appelle "coquille vide" permet de guider l'utilisateur et, par l'examen de règles logiques utilisées par le système, d'examiner les diverses étapes conduisant à une conclusion. On peut modifier les variables principales et effectuer un autre passage machine pour voir dans quelle mesure les résultats sont modifiés. Le système expert, de par sa nature, procure des avantages particuliers dans le domaine de l'évaluation des effets sur l'environnement, opération multi-disciplinaire dans laquelle interviennent des facteurs bio-physiques, socio-économiques et politiques. Les systèmes experts peuvent effectuer un raisonnement qualitatif et un raisonnement fondé sur une connaissance incertaine - facteur important dans de nombreuses évaluations. Le système peut également être relié à des programmes externes, tels que des modèles, qui peuvent traiter des aspects de la situation impliquant une compréhension quantitative.

Les principaux concepts de ces systèmes décrits ci-dessus sont mis en évidence au moyen d'un système permettant de déterminer l'incidence de tunnels sur l'habitat des poissons. On prévoit utiliser le système comme outil d'autoformation dans un cours de niveau supérieur portant sur l'évaluation des effets sur l'environnement.

KNOWLEDGE-BASED OR EXPERT SYSTEMS - A NEW TOOL FOR ENVIRONMENTAL IMPACT ASSESSMENT INSTRUCTION

Alan Gray and Charles Hickman

The School for Resource and Environmental Studies
Dalhousie University, Halifax, N. S. B3H 3E2

BACKGROUND

Knowledge-based systems (KBS) have roots in artificial intelligence (AI) research. Early AI efforts focused on the development of what was termed a "General Problem Solver" - a computer system that would reason from general principles to solve broad classes of problems. This later gave way to the idea that such systems had to be based on knowledge specific to a problem - thus the development of what were termed knowledge-based systems[1]. They may be defined as:

> A computer system that, based on knowledge supplied by human experts, achieves high levels of performance in task areas that, for human beings, require special education and training.[2]

Expert systems (ES) are a sub-set of KBS, those that perform at a level equal to or surpassing that of a human expert. This is still a somewhat controversial concept, which may involve inflated expectations of what a knowledge-based system can accomplish. Both KBS and ES have similar goals, however: the capture of human expertise in a computer-based system, giving the knowledge permanence, and allowing for its wider distribution.

Research and development of KBS were confined for some time to large research institutions such as corporations and universities, due to the high cost of the hardware, software, and technical expertise required. This rapidly changed beginning in the 1980's with the "micro-computer revolution" which resulted in dramatic reductions in the cost of computing power and the consequent development of powerful, relatively low-cost personal computers. Software developers soon followed with a wide array of packages - termed "expert systems shells" - that allow non-programmers to develop specific systems applications, usually by building a knowledge-base of "if...then" rules. As well, AI specific languages such as PROLOG were adapted for use on micro-computers.

Knowledge-based systems differ from conventional computer programs in a number of ways[3]. They are:

- -able to deal with qualitative data
- -able to handle imprecise or incomplete data
- -capable of explaining to the user how conclusions were reached
- -able to interface with the user via "natural language" dialog.

Such characteristics lend themselves to a wide variety of applications in a number of fields, including environmental assessment and management. Their ability to provide explanations to the user makes them a useful tool for training and instruction.

KNOWLEDGE BASED SYSTEMS AND ENVIRONMENTAL IMPACT ASSESSMENT

Generally speaking, environmental impact assessment (EIA) refers to the prediction and assessment of bio-physical impacts likely to result from either a project (e.g. construction of a hydro-electric dam, bridge or power plant) or a more wide spread activity (e.g. pesticide applications, forest management practices). Environmental management refers to more general long-term practices in ecosystem management.

Understanding of the processes involved in both environmental assessment and management is often based on both quantitative and qualitative understanding. As well, data sets may be incomplete. Thus, the characteristics of KBS appear to be ideally suited to such tasks[4]. In the case of areas where there is quantitative, process understanding, KBS may be linked with conventional ecosystem models.

A recent study[5] for the Federal Environmental Assessment Review Office (FEARO) described over forty major KBS and decision support systems in use or under development in this field. Use ranged over such areas as assessment of transportation projects, forest management, integrated pest management, toxic substances management, and risk assessment. Present systems have demonstrated their usefulness, and there are increasing numbers of systems under development. Knowledge based systems promise to provide the conceptual and technical tools to make a substantial improvement in our knowledge management capabilities tc support scientific and management activities.[6]

Knowledge based systems perform a number of functions which make them useful in the environmental decision making process[7,8,9]. They are most useful in situations where there

are routine, but complex, operations that are frequently performed. By capturing expertise and making it permanent, they protect an organization against loss of expertise due to staff turnover. This allows a wider distribution of this knowledge, lessening demands on the human expert's scarce time, and promotes consistency in the decision making process. Due to their nature, such systems offer another significant function: that of training people who do not have the level of expertise of the human expert or experts who contributed to development of the system. Knowledge-based systems fulfill this fuction in two ways: by allowing the user to trace how a conclusion was reached by examining the rules used in the logic process, and by offering explanations of specific concepts and the logic of individual rules.

THE CULVERT INSTALLATION IMPACT SYSTEM

To illustrate the general principles of KBS discussed above, and to demonstrate concepts utilized in EIA, a demonstration prototype KBS was developed at SRES. This system determines the impact of culvert installation on salmonid habitat, and recommends acceptance or rejection of the proposal.

The user enters data interactively requested by the system. Data for a number of parameters are requested: most valuable species in the water-course; number of culverts to be installed; length, diameter, and slope of the culvert; width of overlying road; depth of fill, and area of the watershed. The system is geographically specific. The user is requested to identify the river system, and the nearest upstream and downstream location. This data is used to determine the percentage of spawning habitat that could be affected by such an installation. At present, however, data are available for only one river system.

The culvert installation system was developed using the EXSYS Professional ES shell[10]. Such a shell allows the developer to construct, using an editor, a knowledge base of heuristics or "rules-of-thumb" in the form of "if...then" rules, using a natural language style of input. The shell then builds the KBS, using a programming language (in this case "C"). Some aspects of the culvert system could not be implemented using EXSYS, however. For example, graphic displays are handled by PILOT language programs, and data table lookups by Turbo C. Data may be passed between these external programs and EXSYS.

The user of the culvert system is able to obtain an explanation of the conclusions reached by the system in a number of ways. Firstly, he/she may request a display of the rule involved when a data request is made by the system. The program will also provide a complete tracing of all rules involved in reaching a particular conclusion. This display is colour coded to indicate the logical status of the rule: true, false, or unknown. At each data request, the user may also request further information - an explanation of why the question is being asked, and any terms involved. These latter explanations are provided by the builder of the system during development, and are optional.

A final method of explanation is provided by the report generator that is part of EXSYS. The default output of the system is a table listing the conclusion(s) reached (in this case acceptance, conditional acceptance, or rejection of the culvert installation application) together with an associated probability of the conclusion. This in itself provides little explanatory power. However, the report generator allows the production of a detailed report which explains how the conclusions were reached, and as well outlines conditions which must be altered (e.g. increase in culvert diameter, increase in length, decrease in slope, change in construction time so as to avoid overlap with spawning time) in order for the culvert installation to be approved.

The culvert system offers the user a further option of rerunning the problem while changing one or more of the parameter values. For example, time of construction may be altered and the system rerun without reentering any of the other data. In this way the user can perform varied "what...if" analyses.

Thus, the user of the culvert KBS is provided with access to EIA expertise, and explanations of the reasoning process used to reach conclusions, at the users convenience. The user may request explanation based on his/her own needs, when required. The level of detail offered,however, depends on the thoroughness of the developer.

SUMMARY AND CONCLUSIONS

Knowledge-based systems, either independently or in conjunction with more conventional computer techniques such as models and graphics, have shown themselves to be powerful and useful tools in environmental assessment and management. Their uses are threefold: as problem solving decision aids, as tools for more effectively and efficiently managing complex knowledge, and as instructional aids.

Use of the demonstration prototype culvert KBS has shown it to be an effective instructional tool. SRES has now committed itself to the use of knowledge based systems as learning tools, both as developed at the School, and as obtained from other developers.

ACKNOWLEDGEMENTS

The authors wish to thank Dr. G. Beanlands, Director of SRES, for his helpful guidance in developing the culvert impact system.

REFERENCES

1. DAVIS, J. R., P. T. COMPAGNONI and P. M. NANNINGA. 1987. 'Roles for knowledge-based systems in environmental planning.' Environment and Planning B: Planning and Design, 1987, vol. 14: 239-254.

2. after HAYES-Roth, Frederick, WATERMAN, Donald A., LENAT, Douglas B. 1983. Building Expert Systems. Addison-Wesley Publishing Company, Inc., Reading, Mass., 1983: 444 p.

3. TEXAS INSTRUMENTS INC. 1984. Personal Consultant: Expert System Development Tools. Technical Report.

4. STARFIELD, A. M., BLELOCH, A. L. 1983. 'Expert Systems: an approach to problems in ecological management that are difficult to quantify'. Journal of Environmental Management 16 (1983): 261-268.

5. GRAY, Alan, and Peter STOKOE. 1988. Knowledge-based or Expert Systems and Decision Support Tools for Environmental Assessment and Management: their Potential and Limitations. School for Resource and Environmental Studies, Final Report for the Federal Environmental Assessment Review Office. June, 1988.

6. RAUSCHER, H. Michael. 1987. 'Increasing scientific productivity through better knowledge management.' AI Applications in Natural Resource Management. Vol. 1, No. 2, Fall 1987: 21-31.

7. POLLITZER, Elizabeth and John JENKINS. 1985. 'Expert knowledge, expert systems and commercial interests.' OMEGA International Journal of Management Science 13, 5 (1985): 407-418.

8. HAYES-ROTH, Frederick. 1984. 'The knowledge-based expert system: a tutorial.' Computer (IEEE) September 1984: 11-28.

9. STOCK, Molly. 1987. 'AI and expert systems: an overview.' AI Applications in Natural Resource Management. Vol. 1, No. 1: 9-17.

10. EXSYS, Inc. 1985. EXSYS System Development Package: Users's Manual (+ updates).

PLANNING PROOFS: A PROBLEM SOLVING COACH FOR GEOMETRY

S.H. Greve
Dept. of Computer Science
University of Regina
Regina, SK

This paper describes an interactive intelligent coaching system for high school geometry that addresses the problem of planning the solution to a two-column geometry proof. Many tutoring programs analyze student responses and guide the student through solutions to problems in the domain. The problem, particularly in geometry, is where to start that solution. This coach uses several problem solving approaches. They include: a) analogy--suggesting similar problems; b) construction--adding line segments to the diagram as a means of clarification; c) problem decomposition--breaking the diagram down into all possible geometric figures as a way to show congruences; d) animation--overlaying figures; e) experimentation--testing intermediate hypotheses through proof; f) examples--planning the solution to a similar problem; g) backward-chaining--working backward from the thing to be proved to the givens; and h) brainstorming--listing as many approaches as the student can think of as ways to suggest plans for problem solutions. This coach is used only for planning the solution and is being added to an existing tutor that constructs the proof with the student. The coach, as part of a larger system, has the student model at its disposal which allows it to check the student's previously demonstrated use of geometric concepts. The geometry tutor also contains an expert system that generates the proof which can then be checked against the student's plan for feasibility. This coach helps inexperienced problem solvers gain the knowledge and practice necessary to develop insight into proofs.

PLANIFICATION DES PREUVES: UN SUPERVISEUR DE RÉSOLUTION DE PROBLÈMES DE GÉOMETRIE

Le présent document décrit un système de supervision interactif intelligent destiné aux cours de géométrie du niveau secondaire. Ce système traite du problème de la planification de la solution d'une preuve géométrique sur deux colonnes. Plusieurs programmes d'enseignement analysent les réponses des étudiants et les guident dans leur démarche de résolution des problèmes. Le problème, surtout en géométrie, est de déterminer le point de départ de la solution. Ce programme de supervision utilise plusieurs méthodes de résolution des problèmes. Il comprend a) l'analogie--suggestion de problèmes semblables, b) la construction--addition de segments de ligne au diagramme comme moyen de précision, c) la décomposition du problème--division du diagramme en toutes ses figures géométriques possibles comme moyen de démontrer les congruités, d) l'animation--recouvrement des figures, e) l'expérimentation--essai d'hypothèses intermédiaires de la preuve, f) les exemples--planification de la solution d'un problème semblable, g) l'enchaînement régressif--régression depuis la preuve à fournir jusqu'aux éléments donnés et h) le remue-méninges--énumeration de toutes les démarches auxquelles l'étudiant peut penser pour arriver à résoudre le problème. Ce programme de supervision n'est utilisé que pour planifier la solution et il est ajouté à un guide déjà existant qui élabore la preuve avec l'étudiant. À l'intérieur d'un système plus vaste, le programme superviseur dispose du modèle de l'étudiant et peut vérifier l'utilisation des concepts géométriques déjà faite par l'étudiant. Le programme de supervision de géométrie contient également un système expert qui élabore la preuve qui peut ensuite être comparée au plan de faisabilité de l'étudiant. Ce programme aide les étudiants qui n'ont pas d'expérience dans le domaine de la résolution de problèmes à acquérir les connaissances et la pratique qui leur permettront de traduire leur intuition en preuves.

PLANNING PROOFS: A PROBLEM SOLVING COACH FOR GEOMETRY

Steven H. Greve

Dept. of Computer Science
University of Regina
Regina, Sk.

1. Introduction

Intelligent tutoring systems fall into two categories - programs that teach students and programs that coach students. The teaching programs tend to incorporate tutoring strategies and analyze student responses post hoc. Coaching systems tend to guide the student in real time, usually without prescribed curriculum. This paper describes a problem solving coach that assists students when planning a geometry congruence proof. (For more on planning, see Miller, 1982; Genesereth, 1982; Peachey and McCalla, 1986) This coach is the front end of a larger system that helps the student in developing the proof. The larger system (Greve, 1988; Greve, 1989) is a real time program that consists of an expert system and the tutor/coach. The expert system is used to solve the proof and supply the proof steps to the tutor. The tutor takes the solution and helps the student develop a proof. The tutor maintains a large data structure, the student model, that attempts to track which geometry concepts the student is strong on and where the student is weak. The data structure is organized as a hierarchy of nodes representing these concepts, with the fundamental concepts, lines, points, and angles at the root of the tree. Each of these nodes also contains example templates. These templates are used by the example generator to do demonstrations using a graphic interface during problem solving and proof planning.

The front-end coach described in this paper consists of eight modules, each using a different approach to the initial stages of problem solving and proof planning. The eight modules are: analogy, construction, problem decomposition, animation, experimentation, examples, backward chaining, and brainstorming. The following section describes these modules in detail.

2. Methods

To solve a traditional two-column geometry proof the student must know two things. First, the primary goal and any intermediate sub-goals (other triangle congruences) required to develop the proof and second, the student must have the necessary techniques to start the proof and develop a simple proof plan. The modules in this coach offer the student the tools to enhance their problem solving skills and develop that plan.

2.1 Analogy

As part of the student model the coach maintains a complete list of problems the student has solved as well as problem templates used to create new proof problems on the fly. This affords the problem solving coach the opportunity to suggest analogous solutions that have been previously solved by the student if they are relevant to the current problem. This approach is used to jog the student's memory in an attempt to get the proof started.

2.2 Construction

Construction has been a standard problem solving technique in geometry. For example, the problem in figure 1A was suggested by Euclid and is often referred to as the "Bridge of Fools". Euclid was trying to prove that the base angles of an isoceles triangle were congruent. He did it by adding the line segments at the

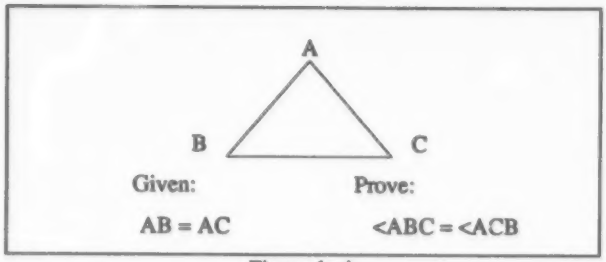


Figure 1 -A

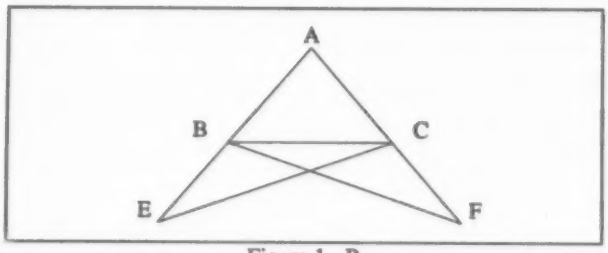


Figure 1 - B

bottom of figure 1-B. He could the show that triangles AEC and AFB were congruent by SAS, and consequently angle AEC is congruent to angle AFB and segment EC equals FB. The remainder of the proof is straightforward.

This module lets the student add lines to the diagram as a means of clarifying the goal or as a way to start the proof. The coach will also suggest constructions if the student is doing one of the built-in proofs or an example generated using the problem templates.

2.3 Problem Decomposition and Animation

One advantage of using a computer as a teaching medium is its ability to respond interactively. Using the problem decomposition module, the student can have the coach separate out all of the geometric figures in the problem. In figure 2A, the original problem is proposed. If the student has difficulty in visualizing the various triangles, the module decomposes it as in figure 2B (only partially decomposed). If it is a particularly complex problem this module can animate the various triangles, angles, and segments. By overlaying the components, the student can get more of a feel for congruences than might be otherwise possible from a static display of the

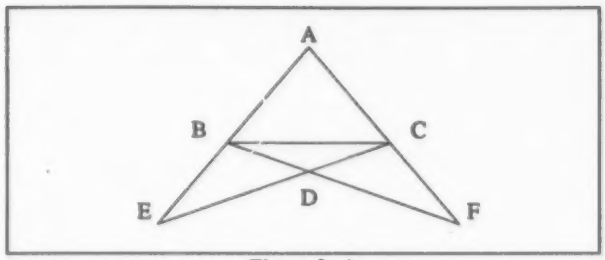


Figure 2 -A

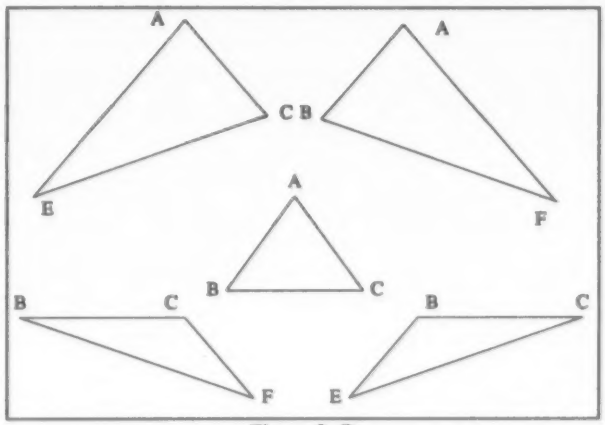


Figure 2 -B

diagram.

2.5 Experimentation

The experimentation module allows the student to test hypotheses by doing partial proofs. For example, the proof in figure 3 requires as an intermediate step that the student produce a triangle congruence. The student could use the experimentation module to find one or more correct solutions. The module allows the student to take part of a proof and work it in isolation.

2.6 Examples

The process of learning to generate longer, more complicated geometry proofs is necessarily built on simpler, shorter proofs. The current system keeps all of the solutions generated by the student and can refer back to one of them as an example of a similar problem to the one they are attempting now if it is applicable.

2.7 Backward Chaining

This module allows the student to work backward from the goal to the given problem conditions. Faced with a long proof and not knowing where to begin, working backward can be a very effective technique. Using the graphic interface the module can visually connect the final goal to all possible ways of attaining that goal. This is a reasonable approach in geometry proof generation as the search space is quite small.

2.8 Brainstorming

If the student does not have a specific idea on where to start a proof, the brainstorming module allows the student to list all the opening moves he/she can think of. Since the system already has the proof solution, this module can check the student's proposed steps against the proof. If the student has listed the right opening move, the module will confirm it. If the student has not listed the correct opening, the module calls the coach to check the student model to see if the student is weak on this concept and if so, offer remediation.

3 Conclusions

This coach is used only for helping the student with problem solving strategies and planning the solution and has been added to an existing tutor that constructs the proof with the student. The coach, as part of a larger system, has the student model at its disposal which allows it to check the student's previously demonstrated use of geometric concepts as well as the expert system that generates the proof which checks the student's plan for feasibility. This coach helps inexperienced problem solvers gain the knowledge and practice necessary to develop insight into proofs.

4. References

Genesereth, M. R (1982). The Role of Plans in Intelligent Teaching Systems. In D. Sleeman and J. S. Brown (Eds.), *Intelligent Tutoring Systems*, (pp. 137-156). New York: Academic Press.

Greve, S. H. (1988) Performance Considerations in the Design and Implementation of Intelligent Tutoring Systems. *Proceedings of the Third International Conference on Symbolic and Logic Computing*.

Greve, S. H. (1989). A Real-Time Coaching Environment for Triangle Congruence Proofs. *Proceedings of the Second International Conference on Computer-Assisted Learning*. Springer-Verlag.

Miller, M. L. (1982). A Structured Planning and Debugging Environment for Elementary Programming. In D. Sleeman and J. S. Brown (Eds.), *Intelligent Tutoring Systems*, (pp. 119-136). New York: Academic Press.

Peachey, D. R. and G. I. McCalla. (1986). Using Planning techniques in Intelligent Tutoring Systems. *International Journal of Man-Machine Studies*. Vol 24, pp.77 - 98.

Polya, G (1957). How To Solve It. New York: Doubleday.

G: CRÉATION D'UN DIDACTICIEL PAR OBJETS

M.W.F. Meurrens
Computer Graphics and Scientific Applications
Bruxelles, Belgium

G est un système d'apprentissage intelligemment assisté par ordinateur fondé sur plusieurs concepts de l'I.A. Contrairement à d'autres systèmes, G n'est pas expert dans un domaine particulier mais plutôt un expert pédagogue offrant une "coquille" (shell) à remplir par l'auteur.

En mode auteur, G permet la concection d'un didacticiel sous forme d'objets liés entre eux par des relations de nature pédagogique. Aucun scénario n'est prédéfini. Les objets sont notamment des unités de connaissance, des hypertextes, des aptitudes d'élève, des croyances erronées, des stratégies pédagogiques, des procédés tutoriels, des réactions prévisibles, des synthèses, des bilans, etc.

En mode élève, G élabore et tient à jour un modèle de l'élève fondé sur les niveaux de maitrise et de pertinence de chacun des objets. Il utilise ce modèle pour construire dynamiquement le dialogue.

G a considérablement évolué depuis sa première présentation (Ottawa, 1986): nouveaux types d'objets, interface homme-machine convivial, contrôle de périphériques, etc.

G: CREATION OF COURSEWARE USING OBJECTS

G is an intelligent computer-aided learning system based on several AI concepts. Unlike other systems, G is not an expert in a specific domain, but rather an educational expert that provides a shell to be filled by the author.

In author mode, G can be used to design a courseware in the form of pedagogically interrelated objects. There is no predefined script. The objects may be knowledge units, hypertexts, student aptitudes, erroneous beliefs, pedagogical strategies, tutoring procedures, foreseeable reactions, syntheses, assessments and so forth.

In student mode, G develops and updates a student model based on the level of mastery and pertinence of each of the objects. It uses this model to construct a dialogue dynamically.

G has evolved considerably since its introduction (Ottawa, 1986), with such improvements as new types of object, user-friendly interface and control of peripheral units.

G: création d'un didacticiel par objets.

Ir. Marc W.F. MEURRENS

Computer Graphics and Scientific Applications
51, rue des moutons (tél. + 32 2 343 4360)
B-1180-Bruxelles (Belgique)

Introduction

Le logiciel G est un système d'"apprentissage intelligemment assisté par ordinateur". Il a été présenté pour la première fois à la communauté scientifique à Ottawa, en 1986. Dès sa version 1.00, G mettait en oeuvre trois concepts essentiels:

- un système d'E.A.O. intelligent doit abandonner l'idée d'un scénario prédéfini par l'auteur et permettre à l'élève un cheminement personnel au sein de la matière enseignée;
- un modèle adéquat de l'élève est absolument nécessaire pour réaliser une individualisation de l'enseignement ou, plus généralement, du dialogue;
- l'auteur d'un didacticiel doit pouvoir décrire, sous forme d'objets, la matière à enseigner et les procédés à utiliser. La description de ces objets met en oeuvre des assertions pédagoqiques et non des instructions de programmation.

Depuis, le système G a évolué (la version 2.00 de G est disponible depuis mai 1989 dans l'environnement MS.DOS sur matériel 80286 ou 80386 avec disque dur) et intègre des concepts venus de l'E.A.O. classique, de l'E.I.A.O. ou, plus généralement de l'I.A., et, enfin, des systèmes documentaires intelligents du type "hypertextes".

G inclut un interface auteur très convivial, un interface élève fondé, soit, sur le language naturel, soit, sur l'usage du curseur.

Système-auteur

G reprend à son compte l'économie de temps qu'un système auteur permet de réaliser par rapport aux languages de programmation et aux langages de l'I.A.

Son interface auteur amical est complété par la possibilité d'introduite les données textuelles massives à l'aide d'un traitement texte habituel.

Un interface avec le système d'opération MS.DOS permet l'utilisation de fonctionnalités non explicites dans G.

La définition d'"objets" remplace la programmation.

Nous verrons plus loin que G est aussi un système élève car, de par sa philosophie même, il ne génère pas un produit compilé capable de reproduire un scénario. Au contraire, G est présent en mode de travail élève afin de déterminer dynamiquement la poursuite du dialogue, sur base, notamment, du modèle de l élève.

L'E.I.A.O.

La structure classique d'un système d'E.I.A.O. (ou d'un "Intelligent Tutoring System"), dédié à un domaine particulier, comporte une représentation du domaine, une représentation de l'apprentissage, un dialogueur et un interface en language naturel.

G se rapproche de cette structure par la nette distinction entre domaine enseigné, approche pédagogique, interface élève-système. Bien entendu, G met en place avec l'élève un dialogue à initiative partagée.

G 'écarte toutefois de l'E.I.A.O. classique en ce qu'il n'est pas un système expert du domaine enseigné et qu'il ne peut donc pas générer et résoudre lui-même des problèmes propres au domaine.

G est, plutôt, un système expert pédagogue qui offre à l'auteur un moule ("shell") qu'il remplira avec les éléments du domaine et ses stratégies éducatives propres.

Le point de vue opérationnel

G se veut, avant tout, un outil disponible. Nous nous sommes donc astreints à n'implémenter que les idées susceptibles de conserver à l'application un caractère opérationnel dans un environnement éducatif ou de formation.

Implémenter des concepts comme la logique floue et un modèle relativement sophistiqué de l'élève sur un ordinateur personnel des premières générations (genre XT) est une gagure qui conduit à des performances décevantes.

Il n'est pas inintéressant de concevoir des recherches avancées avec l'aide d'équipements notoirement liés à l'intelligence articielle.

Pour demeurer accessible au monde de la formation et tirer profit des expériences réelles, le système G est opérationnel sur machines 80286 et 80386 avec disque dur, dans l'environnement MS.DOS. Un coprocesseur mathématique n'est pas indispensable. Les performances quantitatives et qualitatives de G augmentent avec une mémoire supérieure à 640 Kb.

G fait partie des rares systèmes de l'E.I.A.O. qui sont aujourd'hui opérationnels tant pour la recherche pédagogique que pour la production de didacticiels.

Utilisation d'une logique floue.

G utilise, à plusieurs stades du déroulement du dialogue avec l'élève, des emprunts à la logique floue. C'est ce qui nous fait écrire que les choix posés par G sont plus plausibles que certainement exacts.

G utilise également, lorsque plusieurs directions sont grosso modo semblablement plausibles, des algorithmes aléatoires.

Création d'un didacticiel

La création d'un didacticiel sous G ne comporte aucune programmation. L'effort de programmation, de mise en écran, etc -bien connu dans les systèmes d'E.A.O. classique- est ici remplacé par une modélisation de la matière à enseigner et des procédés pédagogiques à mettre en oeuvre.

Cette modélisation se réalise sous forme d'"objets". Les objets ont une définition précisée par l'auteur du didacticiel. En cours de dialogue avec l'élève, G choisit judicieusement, à chaque étape, l'objet dont le déroulement (ou "éxécution") semble le plus plausible.

Par sa structure modulaire, un didacticiel sous G peut être réalisé de façon incrémentale et par des équipes (par exemple: spécialiste du domaine et pédagogues).

Ainsi, il arrivera souvent qu'une première version d'un didacticiel comprenne essentiellement une modélisation des connaissances, tandis que -après une première phase d'expérimentation mettant en évidence les explorations spontanées de l'élève et ses limites- la modélisation des procédés pédagogiques destinés à completer l'initiative de l'élève par des procédés pédagogiques appropriés -questions, bilans, synthèses, etc- fasse l'objet d'une deuxième phase. Il faut noter que G est son propre outil de collecte de données en vue de l'amélioration d'un didacticiel existant.

Divers types d'objets

Plusieurs types d'objets sont prévus par G qui laisse à l'auteur le soin d'en développer d'autres pour les besoins de ses propres recherches.

Parmi les objets proposés par G, citons:

(1) pour la modélisation de la matière et plus généralement du contenu de la formation, les objets C (éléments de connaissance), les synthèses, les savoir faire de l'élève, etc;

(2) pour la modélisation de l'apprentissage, les stratégies pédagogiques de l'auteur, les modules d'interventions, les réactions à des réponses possibles, les bilans, les exemples, etc;

(3) pour un affinement du modèle de l'élève, les croyances possibles (exacts ou erronées) de l'élève, etc;

(4) divers objets de service: leçon, groupe d'élèves, boite aux lettres, etc.

Paramétrisation

Le système G inclut plusieurs centaines de paramètres permettant à un auteur de didacticiel de moduler à sa convenance, voir d'inverser, les règles pédagogiques comprises dans le système. D'une manière générale, les valeurs par défaut sont utilisables telles quelles.

Néanmoins, lorsque G est utilisé en sa qualité d'outil de la recherche et de l'expérimentation en éducation et formation, ou lorsque le didacticiel réalisé vise une cible très particulière, l'auteur complète la réalisation de son didacticiel par la mise au point de divers paramètres.

Structure d'un objet G

(1) La définition interne d'un objet décrit la séquence des événements -visibles ou sensibles pour l'élève- qui prendront place lorsque l'objet sera choisi et exécuté. Parmi ces événements, notons l'apparition d'un texte à l'écran et l'exécution d'une commande de MS.DOS. Cette dernière possibilité ouvre la voie à l'univers multimédia.

(2) La définition de l'objet implique une description des relations qui unissent les objets en terme de pertinence et d'acquisition. Par exemple, le déroulement d'un objet A peut avoir comme effet l'augmentation du niveau d'acquisition (par l'élève) du concept contenu dans A, ainsi qu'une augmentation des concepts B et C qui sont aussi repris dans A. Ou encore, le déroulement d'un objet D devrait être suivi tôt ou tard par celui de l'objet E qui le complète utilement: on dit que D augmente la "pertinence" de E.

Dans la création d'un didacticiel sous G, c'est la description des relations entre objets qui est certainement la phase la plus riche en réflexions.

(3) La définition externe d'un objet décrit les règles (avec un mécanisme de détermination d'un poids) et les conditions qui doivent être satisfaites pour qu'un objet soit candidat au déroulement.

Les conditions portent sur l'intervention de l'élève (question ou réponse), sur l'environnement ou le contexte récent, et, bien entendu, sur le modèle de l'élève.

(4) La définition dynamique d'un objet est l'enregistrement des événements survenus, en session élève, autour et à propos de cet objet: intervention de l'élève, mise à jour du modèle, etc.

Cette définition dynamique fait de G un outil précieux d'expérimentation.

G-texte

Une des composantes d'un objet G peut, à partir de la version 2.00, être un "G-texte' -concept voisin bien que différent de l'hypertexte.

Les textes, et leurs éléments structurels, peuvent être introduits par un auteur, soit, directement avec G, soit, fort simplement, avec un quelconque traitement de texte dont un fichier Ascii sera ensuite traité par G.

Lors du dialogue avec l'élève, les textes apparaissent dans des fenêtres empilées que l'élève manipule par déplacement du curseur ou par d'autres moyens à sa disposition.

Trois éléments essentiels différencient toutefois l'usage des G-textes de celui des hypertextes.

(1) Chaque G-texte fait partie de la définition plus générale d'un objet G. Il est vrai qu'un objet G pourrait ne contenir comme unique élément que un G-texte. Dans ce cas, l'élève verrait à l'écran un empilement de fenêtres comme avec un hypertexte classique. Toutefois, un objet G peut contenir, outre un G-texte, d'autres "instructions", par exemple, des commandes du MS.DOS permettant l'appel à toute procédure externe (graphiques, multimédia divers, etc).

Un objet G peut aussi ne contenir aucun élément

exécutable et visible pour l'élève. Il peut n'être qu'un "environnement" au sein du quel, en fonction de règles et de conditions portant sur le modèle de l'élève, d'autres objets - d'autres G-textes- seront choisis.

Dès lors, l'appel, par le curseur de l'élève, à une poursuite du dialogue dans une direction donnée peut conduire à un déroulement beaucoup plus complexe qu'un simple empilement.

(2) La structure d'un G-texte permet une double possibilité de "branchement". D'une part, on peut prévoir, assez classiquement, qu'à une action de l'élève corresponde l'appel d'un nouveau G-texte (ou plus précisément d'un nouvel objet); d'autre part, on peut prévoir que l'action de l'élève produise simplement une élévation du niveau de "pertinence" d'un nouvel objet sans que, pour autant, cet objet soit immédiatement choisi. Dans ce dernier cas, l'objet concerné aura, en fait, plus de chance d'être ultérieurement déroulé.

(3) Les G-textes faisant partie d'objet G comme les autres, ils participent au modèle de l'élève (voir infra). Ils ont un niveau d'acquisition et un niveau de pertinence tenus à jour.; leur déroulement peut avoir une influence sur les niveaux d'autres objets; ils peuvent être choisis automatiquement (sans intervention directe de l'élève) sur base, notamment, de leur niveau de pertinence.

Du point de vue de l'implémentation, G met en oeuvre divers dispositifs anti-boucle et anti-saturation de mémoire. Placé face à un G-texte, l'élève dispose notamment d'une clef pour obtenir une aide générale sur le texte et de cinq clefs par ligne (l'auteur d'un didacticiel nommera ces clefs comme il l'entend : aide, précisions, rappel, exemple, etc ...). La longueur d'un G-texte, bien qu'optimalement d'une dizaine de lignes, n'est pas limitée par la dimension verticale de l'écran.

Disponibles depuis peu, mais répondant à la demande de nombreux utilisateurs de G, les G-textes donnent déjà lieu à plusieurs expériences: usage dans un didacticiel, gestion d'une bibliographie intelligente, maitrise d'un vocabulaire scientifique, applications aux QCM nuancés.

Le modèle de l'apprenant

Le modèle de l'apprenant est la clef de voûte du système.

Une approche simpliste (mais implémentable sur des ordinateurs plus anciens et moins performants) nous avait conduit, dans des travaux antérieurs, à retenir par une valeur booléenne le fait qu'un texte avait, ou non, été déroulé à l'élève.

L'expérience et la réflexion conduise à un modèle qui mémorise, pour chacun des objets constitutifs d'un didacticiel, pour un élève donné et à un instant donné, un "niveau d'acquisition" et un "niveau de pertinence". Ces niveaux sont des valeurs comprises entre 0 et 100 points ce qui rend le modèle nuancé (fuzzy). Cette approche est aujourd'hui réalisable sur des équipements standards.

De nouvelles réflexions -et des équipements plus puissants- conduiront sans doute demain à raffiner ce modèle, notamment par une mesure de la confiance du niveau enregistré.

Le modèle se complète par des règles (paramétrables par l'auteur) propres au système. Par exemple, G accompagne toute diminution du niveau d'acquisition d'un objet quelconque par une forte augmentation du niveau de pertinence de cet objet -ceci afin de corriger le plus rapidement possible un élève. On peut imaginer qu'un auteur inverse la règle, préferant reléguer dans l'oubli les éléments difficiles pour l'élève.

La mise à jour du modèle de l'élève se fait dynamiquement en cours de dialogue avec l'apprenant. Le modèle individuel est enregistré (sous forme d'un objet) en vue d'une session ultérieure ou d'un examen par un enseignant.

Dialogue à initiative partagée

Le système auteur G ne produit pas un didacticiel exécutable indépendamment. G est aussi le système élève qui gère le dialogue.

Le dialogue avec l'élève permet à ce dernier de prendre l'initiative chaque fois qu'il le souhaite. Les valeurs par défaut des paramètres du système G accordent, en effet, un poids plus élevé aux règles de déroulement correspondant à une question posée par l'élève qu'à une question posée par le système.

L'élève ne doit connaitre que peu de conventions: l'usage, en guise d'aides, de quelques touches fonctions et la manipulation, désormais populaire, d'un curseur.

Placé face à un G-texte, l'élève manifeste ses desiderata, essentiellement, par une action sur le curseur.

Placé dans une position plus générale du dialogue, l'élève exprime ses réponses, ses questions ou ses commentaires en "language naturel approximatif" en les dactylographiant au clavier.

Il y a alternance entre les interventions de G et celles de l'élève. Chaque fois que son tour vient, G choisit, parmi tous les objets du didacticiel, celui dont le déroulement est le plus plausible.

Analyseur morphologique de questions et réponses

G n'inclut pas, à ce stade, d'analyseur syntaxique et, encore moins, sémantique. La recherche est toujours en cours à ce sujet.

G s'inspire, en les complètant par une logique floue (fuzzy), des méthodes du "pattern matching".

L'analyseur morphologique de G prend en charge les variations orthographiques.

La référence par l'auteur à l'analyseur se fait au moment de la définition des règles de sélection des objets (définition externe). Si, par exemple, l'auteur veut conditionner le déroulement d'un objet à la présence de l'expression "poisson rouge" dans une intervention de l'élève, la condition prendra une forme comme celle-ci : "si l'intervention de l'élève, analysée avec une tolérance de <10> points, contient une expression qui ressemble, à <85> points, à l'expression <poisson rouge>". On voit, par cet exemple, qu'on est loin, ici, des techniques utilisées dans les langages auteurs de l'E.A.O. classique.

Bien entendu, les variables peuvent être fixées à des niveaux tels qu'une orthographe parfaite puisse être exigée et que des erreurs puissent être détectées. D'autre part, des fonctionnalités utilitaires permettent de redéfinir un

analyseur pour une langue étrangère ou un public cible particulier.

Conclusions

G enlève à l'auteur les soucis de la programmation mais il lui retire la maitrise du déroulement du dialogue. Il lui impose une réflexion sur le contenu, sur les liens entre les éléments du domaine, sur les stratégies et les méthodes pédagogiques.

G offre à l'élève un univers d'exploration combiné à une assistance tutorielle en cas de défaillance.

G est, pour le chercheur, un outil de réflexion et d'expérimentation.

CONCEPT CLARIFICATION AND VISUAL REPRESENTATION FOR A KNOWLEDGE ENGINEERING ADVISORY SYSTEM

D. Meilleur, P.D. Mitchell
Centre for System Research and Knowledge Engineering
Concordia University
Montreal, PQ

How can general knowledge be represented for different purposes? This paper presents R & D undertaken to develop a visual representation system that fills the gap between sophisticated mathematical representations and descriptive prose. Our Knowledge Engineering Advisory System allows users to classify abstract knowledge in different domains. The user may select different levels of detail for concept representation and be directed to a Help editor which structures the topics and relations for the chosen level of detail. Thus, at one level, the user identifies central concepts that need to be included but does not specify their relationships. At another level, he can establish hierarchical link structures, regroup, define overlaps, or develop precise definition of interrelated concepts. The user may begin the knowledge representation process at any of these levels of complexity and shift to another as required. Illustrations will be provided.

Our Advisory System permits one to compose different kinds of knowledge representation for different purposes (e.g. designing text, CAL, ITS). Or a student can clarify his knowledge of a subject at different levels of detail. Our own research aim is to integrate it into an authoring aid for an ITS.

SYSTÈME GRAPHIQUE D'AIDE À LA REPRÉSENTATION DES CONNAISSANCES

La représentation des connaissances au moyen des réseaux de concepts repose souvent sur des techniques improvisées. Or, cette façon de procéder peut créer de la confusion chez les utilisateurs qui ne savent plus comment interpréter le symbolisme de la représentation. Ainsi, ce qui devait aider à la compréhension se transforme trop souvent en obstacle. Ce texte présente les résultats d'un travail de recherche et développement portant sur l'élaboration d'un système d'aide à la représentation des connaissances. Le système est basé sur une typologie des relations entre concepts mise au point par les auteurs. Cette typologie dépasse le simple système de références pour atteindre le niveau logique de la théorie naïve des ensembles.

Le système permet de représenter graphiquement les réseaux de connaissances. On peut l'utiliser dans divers modes allant de la simple identification des concepts fondamentaux à des structures complexes intégrant les relations logiques entre concepts. L'utilisateur peut, en tout temps, passer d'un mode d'utilisation à un autre. De cette façon le système s'adapte à de nombreuses fonctions telles la conception de textes didactiques ou l'élaboration de cours assistés par ordinateur ou de tuteurs intelligents.

CONCEPT CLARIFICATION AND VISUAL REPRESENTATION FOR A KNOWLEDGE ENGINEERING ADVISORY SYSTEM

Diane Meilleur and P. David Mitchell

Centre for System Research and Knowledge Engineering
Graduate Programme in Educational Technology
Concordia University, Montreal H3G 1M8

SUMMARY

How can general knowledge be represented for different purposes of such diverse users as instructional designers, teachers, students and managers? This paper presents R & D undertake to develop a visual representation system intended to fill the gap between sophisticated mathematical representations and descriptive prose. The resulting Knowledge Engineering Advisory System should allow users to classify abstract knowledge in different domains.

The user may select different levels of detail for concept representation and be directed to a Help editor which structures the topics and relations for the chosen level of detail. Thus at one level the user identifies central concepts that need to be included but does not specify their relationships. At another level the Advisory System establishes hierarchical link structures; similarly we can regroup, define overlaps, or develop a detailed conceptual entailment structure for precise definition of inter-related concepts.

The user may begin the knowledge representation process at any of these levels of complexity and shift to another as required.

This Advisory System should help the instructional designer to compose different kinds of knowledge representation for different purposes (e.g text, CAL, ITS). Equally a student can use it to clarify his knowledge of a subject at different levels of detail. Our own research aim is to integrate it into an authoring aid for an Intelligent Tutoring System.

TEACHING AND LEARNING AS SHARING OF A KNOWLEDGE STRUCTURE

Instructional design, teaching and learning (like management and planning) demand the organization of knowledge, old and new, into a new representation. The precise mechanisms for doing so are elusive but it is clear tht people construct their representations in fundamentally different ways when asked to study something. The regulatory model used determines how they represent, and how well they understand, what they study.

Teaching is more than telling. Instruction is concerned with "implanting of a structure in a mind" and "results from a search for meaning in the structure" (Searles, 1967, p. 24). Instructional designers often mislead themselves into believing that knowledge can be structured and presented in one "best" way and, moreover, that the student can be viewed as a system for both acquiring this particular model of the subject matter or skill and reproducing it upon demand. This is understandable insofar as one may be addressing a specific motor skill or principle (e.g. Ohm's law).

Even so, the ability to define or state a principle does not mean that the student sees the world as the expert does. Probably because principles are easily verbalized we find it easy to accept ability to recall one as evidence that a student understands it, i.e. shares the same concepual model as the expert or instructinal designer. However the meaning which someone extracts from an information source is a function of how the presentation is perceived and organized by the learner as well as by the would-be communicator. And this, in turn, depends upon the purpose for which the learner chooses to watch, listen to or read the information, the purpose of the communicator and the context.

Any instructional designer must have and use a model or representation of the subject matter, no matter how inchoate this may be. What might such a model consist of? What have others used?

MULTIPLE REPRESENTATIONS OF KNOWLEDGE

There are qualitatively different ways of modelling the world not just in the sense that the views of biologists, geologists, chemists, fishermen, power engineers and politicians may perceive a hydro-electric development differently but in the sense that each person can choose to shift perspectives and see different "realities" (cf. Marton, 1981). The same reality can be perceived and represented in strikingly different ways. To illustrate, recall that two famous artists, Renoir and Utrillo, painted the same garden but the paintings are remarkably different. Utrillo's emphasizes geometric shadows, converging lines

and an austere wall whereas Renoir's canvas is filled with light and colour that accent flowers, trees and women. If you had seen the original would you say that either represents the garden accurately? What if you examined a photograph and an infra-red photograph of the garden? Or some other representation? Similarly different models of the political situation in Nicaragua may be derived from the same reality. How can we develop an intelligent CAL system to cope with users having different backgrounds, learning styles and purposes?

Knowledge Representation

Educational technology in general and computer aided learning in particular are concerned with the whole process whereby knowledge can be represented, preserved, organized and made available in a form suitable for someone seeking understanding of that subject matter. We are concerned here with one of the key activities in this endeavour, knowledge representation -- in any form from prose to hypermedia.

How can general knowledge by represented for different purposes of such diverse users as knowledge engineers, instructional designers, teachers, students and managers? Research to answer this question must incorporate findings from many different areas. Despite regular use and encouragement of knowledge representation, relatively little research exists to show its value for students or instructional designers. Knowledge representation is critically important for an Intelligent Tutoring System yet few attempts have been made to integrate a knowledge representation advisory system into an ITS.

Knowledge representation occurs both covertly (in the mind of the researcher, knowledge engineer, subject matter expert or learner) and overtly (in some media open to scrutiny). Educational technologists and other knowledge engineers have spent much effort generating simple representations of what may be worth teaching or learning.

These structures often appear as lists of instructional objectives, sometimes as a hierarchical representation of knowledge, and increasingly as a network of some sort. In doing so, one of the most important aspects of the problem frequently is neglected: a deep and thorough analysis of the conceptual models involved in the mind of the expert and of the learner, as well as a model of the communication and control strategies used to synchronize these and thus to facilitate learning.

In this paper we outline a preliminary attempt to identify and classify simple visually oriented procedures that may be useful in modelling what may be known or done. This research was undertaken to develop a visual representation system intended to fill the gap between sophisticated mathematical representations and descriptive prose. The resulting Knowledge Engineering Advisory System should allow users to classify abstract knowledge about objects, procedures, decisions, processes, concepts or theory.

TOWARD A TAXONOMY OF KNOWLEDGE REPRESENTATION SCHEMES

We undertook the task of preparing a taxonomy to classify different modes of representation using several criteria that permit one to distinguish them. In focussing on visual representations, it is our contention that linear, discursive thought is a special case of a more general cognitive style which can be represented in a variety of ways. Linear prose is one, logic another.

Verbal and Visual Representation

In contrasting language and visual forms, Langer underscored the most radical difference. Visual forms "do not present their constituents successively, but simultaneously, so the relations determining a visual structure are grasped in one act of vision (Langer, 1960, p. 93). Unlike verbal discourse, visual representations are not limited by what the person can remember from the beginning to the end of discourse. The simultaneous juxtaposition of many concepts and their relationships is analogous to a system; the whole is more than the sum of its components considered separately.

Students of engineering and science are familiar with visual representations in the form of chemical formulae, biological processes and schematic diagrams. Often these are essential for an understanding of what is presented in the text or in a lecture. Educational technologists are familiar with slides, graphics, television and other media suitable for representing knowledge. For these purposes, a major virtue of the visual medium is isomorphism. Shapes and procedures can be represented in a one-to-one correspondence in two-dimensional and three-dimensional space (cf. Meredith, 1966).

The visual representation also is suitable for reasoning about these concepts. As Arnheim points out, "This polydimensional space not only yields good thought models of physical objects or events, it also represents isomorphically the dimensions needed for theoretical reasoning" (Arnheim, 1969, p. 232).

Like verbal descriptions, these representations may be considered as partly metaphors, partly analogues, partly symbols. The key question in generating such models is not, "Are they true?" but "Are they useful?"

Static vs. Dynamic Relations

In our taxonomy, the first major distinction is that of time. Some well-known methods of representation involve a temporal dimension. Such time-sensitive models include input-output relations, Critical Path Method, algorithms and certain networks that permit representation of cycles. Moreover a person's conceptual structure changes over time. However the methods of primary importance here are those for which time is not centrally important. This is because the rules governing their construction apparently are not widely known or used in CAL and because the contexts in which they may be used are not always obvious.

Concept Maps

The first group in this taxonomy of knowledge representation methods is the concept map. This is the simplest and most elementary form. It is also the most vague. It is most commonly used when someone wants to explain something, in a spontaneous manner, by showing links between topics or ideas. The only knowledge structure which may be represented is that the author perceives a relationship between one idea and another. A line joining two topics indicates a link.

The meaning engendered by this type of representation often remains vague. Sometimes even the author is uncertain about its utility after awhile.

At this first level of representation is a common form of concept map known as "pattern notes" or a "mind map" (cf. Buzan, 1974; Fields, 1980). Here a central concept is identified which is shown in relation to other concepts that, in turn, may branch to show additional concepts.

This type of map is especially valuable for note-taking during lectures or while reading. Despite its apparent simplicity, its value is far from minor. Habitual users find it possible to retain the essential features of verbal discourse for future use. Conversely, they can use their pattern notes as an outline for writing or speaking.

Conceptual Models

An obvious step from a simple concept map is to use simple symbols, e.g. an arrow, to suggest directionality or a prerequisite concept. Thus a central concept might have arrows leading to it from subsidiary concepts or examples but not from borderline concepts or untruths (cf. Wilson, 1963).

However a much more sophisticated approach seems necessary, one with the potential power of electronic symbols used in a circuit diagram. A more powerful linear graph notation system was developed by Mitchell (1977) to express knowledge structures, to portray a curriculum, to identify instructional modules, to indicate expected learning time, and to represent other information useful for planning. However we need a more comprehensive symbol system.

Such a provisional set of symbols has been offered by Michael Neil. For him, meaning is most important and conceptual models are concerned with the structure or syntax of the knowledge. Illustrative implied relationships include: A is a concept associated with B; A is a property of B; A is a component of B; part of A is converted into B; changes in A will produce changes in B; A is comprised of B and C (cf. Neil, 1970, p. 316).

Hypermedia systems typically contain some architectural features that are compatible with the foregoing but cannot exploit Neil's symbol system fully. Nonetheless they have other features that are potentially useful for representing knowledge, e.g. windows, overviews, and interfaces to control off-line audio-visual storage. We submit that a taxonomy of knowledge representation is also needed for most hypermedia systems. Otherwise they may be little more than an electronic encyclopaedia which the learner may or may not use effectively to construct hiS own knowledge structure..

HYPERMEDIA REPRESENTATIONS

Concordia Course Assembly System and Tutorial Environment

One of the earliest working hypermedia systems is C/CASTE, the front-end of which is a knowledge representation system called THOUGHTSTICKER (Mitchell & Dalkir, 1986). Based on Pask's earlier work (Pask, 1976), this system allows course authors to structure their knowledge in the form of topics and clusters of topics modelled in mesh representations. They then attach, to their meshes, text descriptions, graphic displays, simulations, audiovisual sequences, evaluations and executable models. Coupled with ACME (Mitchell, 1987) the combined system should enable the system to build a representation of the student's understanding of the subject matter, diagnosing various kinds of misconceptions or confusion, even to the point of offering suggestions to the student about selecting further topics that might be learned most readily.

Before representing knowledge as an entailment mesh it is necessary to understand its theoretical foundation (cf. Pask, 1976). For Pask, an entailment mesh representation of knowledge is a directed graph which expresses a prototypical language, Lp. This proto-language is a system that can generate some features thought to underlie natural language. Its

purpose is to render concepts easy to interpret within the framework of Pask's Conversation Theory. In particular it is intended to identify conceptual events which reflect an agreement on the part of two participants that they share a common understanding of a concept.

For understanding to be expressed in the form of an agreement, it is necessary that a concept be perceived as a dynamic process rather than a steady state. To illustrate, if we were studying such inter-related topics as system, system analysis, model and cybernetics, any modification in one's understanding of one of these concepts will modify one's understanding (or potential understanding) of each of the others. The resulting changes in those will influence the first. Such cyclicity is central to Conversation Theory. Note that an agreement on common concepts assumes at least temporary stability and the possibility of memorizing it.

However a stable concept is not merely a reproductive process but also involves production. It involves a repertoire of operations which require using related concepts.

SYSTEMS WITHIN SYSTEMS WITHIN SYSTEMS

A final category involves an extension of hypermedia into recursive systems that become exceedingly complex and impossible to display visually. THOUGHTSTICKER functions at this level and permits a user to regroup clusters of related concepts by changing the focus and context of attention. An example of another recursive representation system, used to model organizations, is Beer's (1986) Viable System Model.

To illustrate how an organizing pattern may change without altering the component concepts, consider this illustration. It is likely that students and faculty, (or labour, management and shareholders) may share the same basic facts, principles, etc. but take very different points of view when interpreting them. What counts is their perception and the organization of concepts which it imposes upon the mesh.

This is exploited in structural communication (Hodgson, 1967) and ACME (Mitchell, 1987) where students are asked to approach a conceptual framework from different perspectives. The student's mesh is then compared with model meshes (representing the different points of view) for evaluation and comment.

FURTHER RESEARCH

The research outlined here is part of an overall research programme intended to develop a Knowledge Engineerig Advisory System that would help to classify abstract knowledge for designers of Intelligent Tutoring Systems and CAL. The intention is that the user may select different levels of detail for concept representation and be directed to a Help editor which structures the topics and relations for the chosen level of detail. Thus at one level the user identifies central concepts that need to be included but does not specify their relationships. At another level the Advisory System establishes hierarchical link structures. Similarly we can regroup, define overlaps, or develop a detailed conceptual entailment structure needed for precise definition of inter-related concepts.

The user may begin the knowledge representation process at any of these levels of complexity and shift to another as required. (Illustrations will be provided at the conference of knowledge representations to cumbersome to include in this paper.)

REFERENCES

Arnheim, Rudolph, Visual Thinking, Berkeley: University of California Press, 1969

Beer, Stafford, Diagnosing the System for Organizations. Toronto: John Wiley & Sons, 1985.

Buzan, Tony, Use Your Head, London: BBC, 1974

Fields, Alan, Pattern Notes, NSPI Journal, October 1980.

Hodgson, A., "Structural programming in education," Systematics, vol.4, pp. 346-370, March 1967.

Langer, Suzanne K., Philosophy in a New Key, Cambridge Mass.: Harvard University Press, 1960

Marton, F., "Studying conceptions of reality" Scand. J. Educ. Res., vol. 25, pp. 159-169, 1981.

Meredith, Patrick, Instruments of Communication, Oxford: Pergamon, 1966.

Mitchell, P.D., ACME: A prototype system for tutorial development and evaluation of understanding, in Proceeding of 1987 I.E.E.E. Conference on Systems Men and Cybernetics.

Mitchell, P.D. & K. Dalkir, "C/CASTE: An artificail intelligence based computer aided learning system", in Computer Assisted Learning, the Next Five Years. Ottawa: National Research Council, 1986.

Neil, M.W., "The Nature, Rules and Construction of Conceptual Models in Higher Education". in Aspects of Ed. Tech. IV, ed. by D. Unwin and J. Leedham, London: Pitman, 1970

Pask, Gordon, Conversation Theory. Amsterdam: Elsevier, 1976.

Searles, John E., A System for Instruction, International Textbook Company, 1967.

Wilson, John. Thinking with Concepts, Cambridge: Cambridge University Press, 1963.

ACKNOWLEDGEMENT

The assistance of grants from the Social Sciences and Humanities Research Council of Canada and from the Quebec Minister of Education's F.C.A.R. are gratefully acknowledged.

COMPUTER ASSISTED EDUCATION/

ÉDUCATION ASSISTÉE PAR ORDINATEUR

RESEARCH AND EVALUATION TECHNIQUES IN COMPUTER ASSISTED LEARNING: A QUASI META-ANALYSIS OF CONTEMPORARY STUDIES

L.H. Sandals
Dept. of Educational Psychology
The University of Calgary
Calgary, AB

In the evaluation of Computer Assisted Learning (CAL), we have seen a movement from the late 60's to the mid 70's of a discipline that used experimental techniques analogous to "John Henry Research" where the man or woman (teacher or trainer) was being pitted against the computer. These studies were usually quite quantitative in nature and usually made use of inferential techniques that more than likely violated many of the assumptions because of the non-randomness of the subjects in addition to small sample sizes and unequal or missing values. Even with these problems, the face validity of the studies were considered to be solid in most academic circles considering the limitation of the hardware. Seidel (1980) began to come to grips with these evaluation issues in his then provocative article "It's 1980: Do You Know Where Your Computer Is?" where he coined the term "Value added Cost to Education". This proposed presentation and article will present a summative look at the Computer Based Learning research by breaking down the 1960's, 70's and 80's by the type of research being carried out and cross tabulated by the techniques being used as well as the changes in research designs caused by the new emerging technologies in computer based learning environments.

TECHNIQUES DE RECHERCHE ET D'ÉVALUATION EN ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR: UNE QUASI MÉTA-ANALYSE D'ÉTUDES CONTEMPORAINES

Au cours de son évolution, l'enseignement assisté par ordinateur (EAO) a connu un certain mouvement qui a prévalu de la fin des années 60 au milieu des années 70. On utilisait alors des techniques expérimentales semblables aux techniques de recherche de John Henry, selon lesquelles on opposait l'enseignant ou l'agent de formation à l'ordinateur. Ces études étaient habituellement de nature bien quantitative et faisaient appel à des méthodes déductives qui fort probablement biaisaient de nombreuses hypothèses étant donné la régularité des sujets, la petitesse des échantillonnages et l'inégalité ou l'absence des valeurs. Même en connaissant les problèmes, les milieux universitaires considéraient ces études comme valides, vu le matériel restreint qui était utilisé. Seidel (1980) commença à s'élever contre ces évaluations et rédigea un article provocateur intitulé :"It's 1980: Do You Know Where Your Computer Is?" (Nous sommes en 1980: Savez-vous où se trouve votre ordinateur?) dans lequel il créa l'expression "Value Added Cost to Education" (coût de la valeur ajoutée pour l'enseignement). Cet exposé vous donne une vue d'ensemble de la recherche en enseignement assisté par ordinateur. On y répartit les années 60, 70 et 80 selon le genre de recherche effectuée et les techniques utilisées, de même que selon les changements entraînés dans les modes de recherche par les nouvelles techniques utilisées pour l'enseignement assisté par ordinateur.

RESEARCH AND EVALUATION TECHNIQUES IN COMPUTER ASSISTED LEARNING: A QUASI META-ANALYSIS OF CONTEMPORARY STUDIES

LAURAN H. SANDALS

DEPARTMENT OF EDUCATIONAL PSYCHOLOGY, THE UNIVERSITY OF CALGARY

Introduction

The use of the computer as an accepted adjunctive and mainline instructional support for both the elementary, secondary, and post secondary teacher and industrial and military trainer has occurred because of the many researchers who have carried out studies in the past twenty five years. This research has taken many forms as documented in the many review studies to follow as well as other scholarly publications and presentations.

This particular paper will focus on what we know as far as accepted research practices in the sciences and social sciences and how this correlates with the actual research that has been carried out in computer based education over the last quarter of a century. This author has in the title and abstract of this manuscript provided a term to define this type of paper as a "Quasi-Metanalysis" of contemporary studies. There have been many Metanalysis' reported in prior research studies in this area and some of them are listed with primary characteristics in Table 1. This paper is a quasi-metanalysis in the fact that the data to be discussed will be a major overview of these Metanalysis unto themselves along with other major reviews in this area. What the author will be most interested in is not the effect sizes themselves but the research designs of the prior studies summarized in the metanalysis including the related independent and dependant variables that were investigated.

Background

Within the above context it can be observed that almost all research in these areas makes use of one of the six major areas of research in the continuum (defined by Hildegard 1964) where we integrate what we know of the principles of learning and how these in turn integrate as principles with research in Computer Based Education. Research using this model can be discussed as follows (Adapted from Decco 1968):

A) Basic Research

Step 1: Research on Learning with no regard for its educational relevance. (e.g. Development of CAL authoring systems. Hardware design such as touch screens and speech digitizers.)

Step 2: Research on learning which is not concerned with educational or training practices but which is more relevant to the end product than step 1 since it deals with human subjects and content that is nearer to that used in educational and training institutions. (e.g. Human subjects involved in the fine tuning of computer hardware and human computer interfaces such as the mouse, power pads, speech devices.) This includes research in human computer interactions but the instructional materials are secondary. Research in this area would also include some of the research in the area of intelligent CAL and the accessibility of digitally stored images and data bases.

Step 3: Research on learning that is relevant because the subjects are in educational or training programs and the material is relevant to the training needs of the subject. The main difference in comparison to the next step, is that the results of this research do not have immediate applicability to existing educational and training programs in a generalized sense. (e.g. Research investigating the effects of sound or color or screen design on the learners performance with educational materials, but without the conceptuality of these materials in an existing curriculum.) In this mode the effects of different types of reinforcement and their respective response rates would be considered. The basic attributes of drills, tutorials, and simulations would be investigated from a more theoretical approach without immediate applicability.

B) Applied Research

Step 4: Research into computer based education usually in a special CAL laboratory or classroom. This may take place in a university even though the subjects are elementary or secondary students or the study may be in a provincial or state research institute. In the case of industrial or government training the study would not be taking place in a realistic field setting. (e.g. The curriculum materials would be well thought out and integrated into different curricula (scopes and sequences) but most of the research at this level would be pilot or exploratory research before larger studies can be implemented in the field.) In the strictest sense this is a total review of the CAL package, the hardware configuration, the related software, and the necessary support materials before multiple field sites are studied.

Step 5: Similar to step 4 but the major difference is that the same configuration and package is tried out in the real world in schools or training centers usually with at least two field test sites. Research at this level starts

focusing on the institutionalization of the computer into the everyday operation of the school or center taking into account such new issues as staff training, entering skills of the individual learner and the reliability of the hardware and software over the long term. (e.g. Same as step 4 except they are implemented directly in the field which amounts to the real world with real time environments.)

Step 6: The research at this level focuses on advocacy and adoptions. Things that are found to work in the computer based education environments in Step 4 and Step 5 have to be packaged for wider use and then have to go through more extensive field evaluations with individuals and sites not originally involved with the research in Steps 1 through Step 5. (e.g. This would mean the extension of a project from perhaps three elementary schools to perhaps fifty in the same school division or fifty spread throughout the province. From the perspective of training programs a project that was used in one military base may, for example, now be tried in three separate bases in three different parts of Canada.

Research in computer based education has taken place at all six of the previous defined steps even though some researchers may always spend most of their research efforts in one step or two, depending on their focus. Many research studies take into account the research of others in steps 1 through 4 and in turn they work at the next level in steps 5 and 6. This is analogus to the Japanese in the production of electronics and cars within steps 5 and 6 where they have taken the research carried out in other countries in steps 1 through 4 and have in turn modified and improved on the research ideas in the development of research and products at steps 4 through 6. This technical leap frogging and extension to step 6 also takes place in many of the computer based education research projects where one research team may take an educational video developed by others and carry out implementation and software development at stages 4-6.

The Teaching Learning Process

All too often we tend to look at research in Computer Based Learning or Education to be an area of study unto itself and not taking into account the more generalized research on teaching such as those in Wittrock (1986). This handbook of research on teaching has many issues that are important in the design of instruction and the teaching process to be used, whether media based techniques are present or not. Shulman (1986) brings this point into focus when discussing "Paradigms and research programs in the study of teaching: A Contemporary Perspective."

The significant body of knowledge on effective teaching from a research perspective discusses the same issues as those that are discussed in Computer Based Learning but on a more wholistic perspective. For example the variables as defined on the continuum are A) PreSage Variables 1. Teacher Format Experience 2. Teacher Training Perspectives 3. Teacher Properties are analogus to the quality of the instructional design, the sophistication of the hardware and software and the richness of the content when developing Computer Assisted Learning Software. B) Context Variables such as 1. Pupil Formation and Experiences 2. Pupil Properties 3. School and Community contexts and 4. classroom context are the same whether the instruction is through a teacher, a teacher using the computer adjunctly or through a lesson where the computer is the main learning mode. Many times in CAL these variables are defined as Pre Entry skills. C) Process Variables include the classroom and classroom behaviour and this variable more than the preceding two is different when addressing the research issues when the computer is and is not imbedded in the educational process. The uniqueness of individual instruction, the active participation and immediate feedback of the Computer Learning environment does provide us with different research variables in comparison to the more typical Teacher Based grouped instruction. D) The Product Variables regularly short term and long term pupil growth are similar whether the computer is or isn't involved in the teaching process. Variables such as 1. Subject Matter Learning 2. Attitudes towards subject 3. Transfer to Growth through other skills and 4. Personality and career skills are all important in both CAL and non CAL research into teaching environments. Thus when Seidel (1980) defines the following variables as effective measures of "effective teaching" in Computer Based Learning one must ask are we discussing something that is truly unique to research in Computer Based Learning? Or are we, as this author proposes, refocusing the more generalized research models for effective teaching, instruction and learning and in turn we have overemphasized the uniqueness of the Computer Based Learning environment. Seidel challenges us with five categories in table 2 which address many of the key dependent variables and product variables which we truly think are important in measuring the effectiveness of Computer Based Learning environments. This author has asterisked and footnoted those that should be considered clearly unique to CAL. As can be observed only 25% may be considered truly different and in fact when we look at research studies on the effects of media in teaching such as Clark and Solomon (1986). None are truly unique unto themselves from a General Research Design perspective on the effectiveness of teaching.

Summary and Conclusions

The growth of computers in the schools has grown to the point where we have seen swings in different research methodologies especially in the period from the late 70's to mid 80's where much of the research tended to move within the Qualitative Domain with many studies being carried out at Step 5 & 6 without the required Qualitative/Quantitative studies at steps 2-4. This was due mainly to the rapid implementation of micros in the school and the home. Some of these Research issues were debated in a special journal on Research in Computer Based Education by Tennyson (1984) where he looks at and discusses the future for Research and the related methodologies for Computer Based Instruction. Clark

(1985) discusses the "Confounding Results in Educational Computing Research" in relationship to the appropriate research strategies for this medium while Clark and Soloman (1986) discuss these effects in media in teaching in general. Other recent research studies which highlight these discrepancies in both Qualitative vs. Quantitative Research Design Issues is found in Hagler and Knowlton (1987) where they discuss the "Invalid Implicit assumptions in CBI Comparison Research" especially in regards to how meaningful independent Instructional variables are with or without the CBI environment. In Reeves (1986) we find that in a review of Research and Evaluation Models for CAL he has found evidence that "Comparative Research Methods using experimental and quasi-experimental paradigms, have failed to accurately assess the effectiveness of Instructional Technology." He concludes by highlighting the need for the development of more exploratory observational studies followed by more effective system-oriented evaluation.

The real questions are whether the issues regarding the effective evolution of Research in Computer Based Learning and Education are that significantly different from those in other disciplines of inquiry. In Special Education, an area this author has considerable experience in, we have seen ongoing debates regarding the effects of Qualitative vs. Quantitative Research and a combination of both methods discussed by Stainback and Stainback (1984), Simpson and Eaves (1985) and Stainback and Stainback (1985). In studies regarding the major Reviews of Research in Education we find Shulman (1986), Walberg (1986), and Giaconnia and Hedges (1987) discussing under the general headings of Research Integration, Synthesis of Research and General Research design issues for Education and Teaching many of the same issues that those in Computer Based Learning have been grappling with for the past 25 years. Bernard (1986) in his editorial comments discusses the following issue "Is Research in New Technology caught in the Same Old Trap?" His assumption is that when we discuss Computer Based Learning environment with and without traditional instructional strategies we tend to mix the proverbial "apples with oranges." His views are that we still try to solve a very complex instructional problem teaching (with the support of technology) with a "once-and-for-all" or more aptly "one-size-fits-all" mentality for Research in Instructional Technology while we should be following the more generalized and complex evaluation models of teaching in general such as those of Shulman (1986) and Walberg (1986).

It is this authors opinion that we as researchers tend to look at Computer Based Learning Environments all too often as a type of instruction completely outside the regular Institutional and Educational Learning environments and thus research in the areas of LOGO or CAL in Basic Skills shows such confounding results in the many metanalysis and review studies in table 1 using the respective Product and Process variables outlined in table 2. This paper is one small step in helping researchers in the area of Computer Based Learning and Education to inturn re-focus on what are the Research and Evaluation issues in our discipline. Is this area so uniquely different from the thousands of research studies carried out in the area of effective instruction with or without the intervention of this technological medium. In conclusion if they are truly different can we detect this difference.

REFERENCES

1. Aiello, Nancy and Wolfle, Lee. A Meta-Analysis of Individualized Instruction in Science, paper presented at the Annual Meeting of the American Educational Research Association, Boston, 1980. ERIC reprint ED3404.

2. Bangert-Drowns, R., Kulik, J. and Kulik, C-L. Effectiveness of Computer-Based Education in Secondary Schools, Journal of Computer-Based Instruction, 12:3, pp. 59-68, 1985.

3. Bernard, R. Is Research in New Technology Caught In the Same Old Trap? Canadian Journal of Educational Communication, vol. 15, no. 3, 147-151, 1986.

4. Burns, P. and Bozeman, W. Computer-Assisted Instruction and Mathematics Achievement: Is There a Relationship? Education Technology, 21, pp. 32-39, 1981.

5. Clark, R. & Salomon, G. Media in Teaching. In M. Wittrock, Editor. Handbook of Research on Teaching, 3rd Edition. New York: MacMillan Publishing Company, 1986.

6. Clark, R. Confounding in educational Computing Research. Journal of Educational Computing Research. 1(2), 1985.

7. De Cecco, J. The Psychology of Learning and Instruction: Educational Psychology. New Jersey: Prentice-Hall Inc., 1968.

8. Edwards, James, Norton, S., Taylor, S., Weiss, M. and Dusseldorp, R. How Effective is CAI? A Review of the Research, Educational Technology, 33, pp. 147-153, 1975.

9. Feldhusen, James and Szabo, Michael. The Advent of the Educational Heart Transplant, Computer-Assisted Instruction: A Review of the Research, Contemporary Education, 43, pp. 265-274, 1969.

10. Giaconia, R.M. and Hedges, L.U. Synthesizing Research Evidence. In M. Dunkin Editor, The International Encyclopedia of Teaching and Teacher Education. Oxford, Toronto: Pergamon Press, 1987.

11. Glass, G.V. Primary, Secondary and Meta-Analysis of Research. Educational Researcher, 52, pp. 3-8, 1977.

12. Hagler, P. & Knowlton, J. Invalid Implicit Assumption in CBI Comparison Research. Journal of Computer Based Instruction, vol. 14, no. 3, 84-88, 1987.

13. Hall, K. Computer Based Education. In H. Mitzel Editor, Encyclopedia of Educational Research, 5th edition. New York: The Free Press, 1982.

14. Hartley, Susan. Meta-Analysis of the Effects of Individually Paced Instruction in Mathematics. Dissertation Abstracts International, 38:7-A, 4003, University Microfilms No. 77-29, 926, 1978.

15. Hilgard, Ernest R. "A Perspective on the Relationship Between Learning Theory and Educational Practices," In Theories of Learning and Instruction, ed. E.R. Hilgard, Part 1 of the 63rd Yrbk. of the National Society for the Study of Education, pp. 402-15. Chicago, University of Chicago Press, 1964.

16. Jamison, D., Suppes, P., and Wells, S. Effectiveness of Alternative Instructional Media, Review of Educational Research, 44, pp. 1-67, 1974.

17. Kulik, C-L and Kulik, J. Effectiveness of Computer-Based Education in Colleges, paper presented at the Annual Meeting of the American Educational Research Association, Chicago, AEDS Journal, 1985.

18. Kulik, J., Kulik, C-L and Bangert-Drowns, R. Effectiveness of Computer-Based Education in Elementary Schools, Computers in Human Behavior, 1, pp. 59-74, 1985.

19. Kulik, J., Kulik, C-L., and Shwalb, B. Effectiveness of Computer-Based Instruction in Adult Education, Journal of Educational Computing Research, 2:2, pp. 235-252, 1986.

20. Niemiec, R., Samson, G., Weinstein, T., and Walberg, H. The Effectiveness of Computer-Based Instruction at the Elementary School Level: A Quantitative Synthesis, paper presented at the Annual Meeting of the American Educational Research Association, Chicago, 1986. AEDS Journal, 1985.

21. Niemiec, R. & Walberg, H. Comparitive Effects of Computer-Assisted Instruction: A Synthesis of Reviews. Journal of Educational Computing Research, Vol. 13, No. 1, 19-37, 1987.

22. Reeves, T. Research and Evaluation Models for the Study of Interactive Video. Journal of Computer Based Instruction, vol. 13, no. 4, 102-106, 1986.

23. Rubincam, I. Frequently cited Authors in the Literature on Computer Applications to Education. Journal of Computer Based Instruction, vol. 14, no. 4, 130-136, 1987.

24. Samson, G., Niemiec, R., Weinstein, T. and Walberg, H. Effects of Computer-Based Instruction on Secondary School Achievement: A Quantitative Synthesis, paper presented at the Annual Meeting of the American Education Research Association, Chicago, 1985. AEDS Journal, 1985.

25. Schmidt, M., Weinstein, T., Niemiec, R. and Walberg, H. Computer-Based Instruction with Exceptional Children: A Meta-Analysis of Research Findings, paper presented at the Annual Meeting of the American Educational Research Association, Chicago, 1985. Journal of Special Education, 19, pp.493-502, 1985-1986.

26. Seidel, R. Its 1980: Do You Know Where Your Computer Is?, Phi Delta Kappan, no. 7, pp. 481-485, 1980.

27. Shulman, L. Paradigms and Research Programs in the Study of Teaching: A Contemporary Perspective. In M. Wittrock, Editor, Handbook of Research on Teaching, 3rd Edition. New York: Macmillian Publishing Company, 1986.

28. Simpson, R. & Eaves, R. Do we need more qualitative research or good research or more good research? A reaction to Stainback and Stainback. Exceptional Children, 4, 325-329, 1985.

29. Smith, M. Research Integration. In H. Mitzel Editor, Encyclopedia of Educational Research, 5th Edition. New York: The Free Press, 1982.

30. Stainback & Stainback. Quantitative and qualitative methodologies: Competitive or complementary? A response to Simpson and Eaves, Exceptional Children, 5(51), 330-334, 1985.

31. Stainback, S. & Stainback, W. Broadening the research perspective in special education. Exceptional Children, 5, 50, 1984.

32. Tennyson, R. Special Issue: Educational Research and Computer Based Instruction. Journal of Computer Based Instruction, vol. 11, no. 1, 1, 1984.

33. Thomas, D. The Effectiveness of Computer-Assisted Instruction in Secondary Schools, AEDS Journal, 12, pp. 103-116, 1979.

34. Vinsonhaler, J. and Bass, R. A Summary of Ten Major Studies on CAI Drill and Practice, Educational Technology, 12, pp. 29-32, 1972.

35. Walberg, H. Synthesis of Research on Teaching. In M. Wittrock, Editor. Handbook of Research on Teaching, 3rd Edition. New York: MacMillian Publishing Company, 1986.

36. Wedman, J. Citation Patterns in the Computer Based Instruction Literature. Journal of Computer Based Instruction, vol. 14, no. 3, 93-96, 1987.

37. Wittrock, M. Editor, Handbook of Research on Teaching, 3rd Edition. New York: MacMillian Publishing Company, 1986.

TABLE 1

Some Major Research Reviews in Computer Based Learning

Studies on the effects of Computer Based Learning and Education Adapted and Modified from Niemiec & Walberg (1987) "Comparative Effects of Computer-Assisted Instruction: A Synthesis of Reads". Journal Education Computer Research, Vol. 3(1), 1961.

AUTHORS	TYPE
Feldhusen & Szabo (1968)	Traditional
Vinsonhaler & Bass (1972)	Quantitative
Jamison, Suppes, & Wells (1974)	Traditional
Edwards, Norton, Weiss, Taylor, & Dusseldorp (1975)	Quantitative
Hartley (1978)	Quantitative
Thomas (1979)	Traditional
Aiello & Wolfle (1980)	Quantitative
Burns & Bozeman (1981)	Quantitative
Hall (1982)	Traditional
Clark (1985)	Traditional
Kulik and Kulik (1986)	Quantitative
Samson, Weinstein, Niemiec & Walberg (1985)	Quantitative
Bangert-Drowns, Kulik & Kulik(1985)	Quantitative
Niemiec, Samson, Weinstein & Walberg (1986)	Quantitative
Kulik, Kulik and Bangert-Drowns (1985)	Quantitative
Schmidt, Weinstein, Niemiec and Walberg (1985)	Quantitative
Kulik, Kulik and Schwalb (1986)	Quantitative
Clark and Solomon (1986)	Traditional
Reeves (1986)	Traditional
Hagler & Knowlton (1987)	Traditional
Wedman (1987)	Traditional
Rubincam (1987)	Traditional
Niemiec & Walberg (1987)	Traditional

Quantitative = usually means some form of meta-analysis.
Traditional = usually means a Literature Review and a review using Research Synthesis Techniques.

TABLE 2

Adapted and modified from Seidel (1980) "It's 1980! Do you know Where Your Computer Is?" Phi Delta Kappan March 1980. pgs. 481-485

Dependent variables that have been identified as significant in Computer Assisted Learning or from the perspective of Effective Teaching Process and Product Variables.

Purpose: Computing Opportunity * (Learning with Computers)
Measures:
* Access to Computing Facilities
* Separate Budget for Instructional Computing
  Time Set Aside for Student and Instructor Use
* Convenient Location(s) of Terminals
  Documentation Available
  Consultation Available

Purpose: Computing Literacy * (Learning About Computers)
Measures:
Courses in
* Understanding of Computers and Impact on Society (Awareness)
* Computer Applications in Own Discipline (Knowledge)
* Computer Science and Programming (Skills)
Percent of Target Population Served
  Students
  Faculty
Achievement in Courses
Value Change Following Course

Purpose: Curricular Enhancement (Learning from Computers)
Measures:
Time
  Average Training Time in Course
  Average Testing Time in Course
  Average Course Time
Achievement (Final Criterion)
Test Results
  Accuracy or Speed Scores
  Gain Scores
  Number of Objectives Passed
  Percent of Students Failed

Purpose: Educational Reform
Measures:
Individual Student Progression
Varieties of Learning Locations
Changes from Credentials to Competencies

Purpose: Cost-Effectiveness
Measures:
Student Efficiency (Achievement/Time, Money)
Cost Avoidance
Student Enrollment
Attendance
Attrition
Other Effectiveness Measures
  Student Attitudes
  Instructor Ratings
  Attrition Rates
  Absentee Rates

* denotes those measures that may be specific only to Computer Learning environments. The others are generic to most forms of research on the effectiveness of teaching.

PROCEDURAL AND DECLARATIVE LANGUAGES: A DISCUSSION FROM AN EDUCATIONAL OUTLOOK

R.M. Bottino
Istituto per la Matematica Applicata
Consiglio Nazionale delle Ricerche
Genova, Italy

The use of programming languages in education is evolved and differentiated during the years. In particular, at secondary school level, various experiences have been developed regarding the introduction and the possible applications of languages of different nature. The choice and the use of a given language rather than another depends from motivations related both to the kind of applications which are to be carried out, and to different educational aims which are linked also to the distinct curricula.

In this paper we examine what advantages and disadvantages the introduction of different programming languages might have for beginner students in the first years of secondary school (age 14-16). Based on some author's research experiences in classrooms, we compare procedural languages and the more recently introduced declarative ones, in particular Prolog, whose increased interest derives from Artificial Intelligence applications. Emphasis is given to the possibilities they offer and to the problems they posed in relation to their teaching and learning and to their use in educational applications. Moreover, we discuss the need for curricula innovations and the problems related to teachers' training, which are induced by the present developments in computer science and their future perspectives.

LANGAGES PROCÉDURAUX ET DÉCLARATIFS : ANALYSE D'UN POINT DE VUE ÉDUCATIF

L'utilisation de langages de programmation dans le secteur de l'éducation a évolué et s'est particularisée au cours des années. Au niveau secondaire, en particulier, on a tenté diverses expériences d'introduction et d'application de langages de nature différente. Le choix et l'utilisation d'un langage donné plutôt qu'un autre dépend des motivations à l'égard du type d'application implantée et des buts éducatifs différents qui sont également liés aux divers programmes.

Dans le présent document, nous examinons les avantages et les inconvénients que pourrait présenter pour les étudiants débutants des premières années du niveau secondaire (14 à 16 ans) l'introduction de différents langages de programmation. En nous appuyant sur certaines expériences d'auteurs dans des salles de cours, nous comparons les langages procéduraux et les plus récents langages déclaratifs, en particulier le Prolog, qui connaît un intérêt accru en raison des applications de l'intelligence artificielle. L'accent est mis sur les possibilités que ces langages offrent et sur les problèmes qu'ils posent aux plans de l'enseignement et de l'apprentissage et pour ce qui est de leur utilisation dans des applications éducatives. Nous parlons, en outre, de la nécessité d'innover en matière de programmes et des problèmes liés à la formation des enseignants qui sont attirés par les récents progrès de l'informatique et par les perspectives que ces derniers ouvrent.

PROCEDURAL AND DECLARATIVE LANGUAGES: A DISCUSSION FROM AN EDUCATIONAL OUTLOOK

Rosa Maria Bottino

Consiglio Nazionale delle Ricerche
Istituto per la Matematica Applicata
Genova- Italy

1. INTRODUCTION

Types of computer applications in education have evolved and differentiated during the years.
This paper focuses on the use of computers in the basic formation in secondary school. From this point of view computers, and more generally computer science, can be seen as both an educational means or as a subject of teaching.
As educational means computers are used for subject matter learning. Numerous applications have been developed for teaching in different subjects. In this field we can find, for example, the various C.A.I. (Computer Assisted Instruction) programs and, more recently, the I.C.A.I. (Intelligent C.A.I.) systems, which apply the techniques and the methods of Artificial Intelligence to educational systems.
As a subject computers are studied to introduce students to some computer science subjects, e.g. programming languages and methods, algorithm development, computer architecture, etc., which are considered important both for a basic knowledge in the field and also for the possibilities they offer to the teaching of other school subjects (e.g. mathematics, linguistics, etc.).
These two ways of using computers in education answer to different educational objectives. On one hand a previously implemented system can be used , without requiring the student or the teacher to have any particular computer science notion, as an expert aid in a given field. On the other hand the teaching of concepts and notions related to the direct programming of a computer gives the possibility to introduce students to a subject which is of great importance to understand present and future reality .
This paper refers to this second aspect of the use of computers in education.
In particular, I examine the possibilities and the problems related to the introduction of different programming languages in the first years of secondary school (age 14-16). Based on some research experiences in classrooms, I compare procedural languages, in particular Pascal, and the more recently introduced declaratives ones, in particular Prolog, whose increased interest derives from Artificial Intelligence applications.
I do not directly consider other educational languages, such as Logo, because usually this language is used at a different school level to that under discussion.
The introduction of some computer science elements to students can be performed following two different orientations: introduction of these concepts as a self standing teaching subject or inside the teaching of another school subject.
These two possibilities depend on school organization and also on different, but in some cases complementary, educational goals.
In the first case there is the desire to give students some concepts about computer science because they are considered important for their basic formation.
In the second situation the focus is more on the use of computers as a tool towards the acquisition of knowledge, on the assumption that more powerful ways of representation in the mind of students can be achieved.
So, in this respect, computers are seen as a means to enhance the learning of given topics.

2. PROCEDURAL AND DECLARATIVE PROGRAMMING

Given the problem of finding the maximum of a set of numbers, consider the two following programs:

a)
```
max([], undefined).
max([M|[],M).
max([H|T],M) :- max(T,M), M>H.
max([H|T],H) :- max(T,M), M< H.
/* the first clause may be deleted which will result in "fail" if
max is called with an empty list */
```

b)
```
Program max;
 Type val = array [1..n] of real;
 Var num: val;
     max: real;
      k,i,n: integer;
 procedure readdata;
  begin
    readln(n);
    i := i+1;
    repeat
     readln (num(i));
     i:= j+1;
    until i = n
  end;
 procedure findmax;
  begin
    k := 1
    max := num(k);
    repeat
     k := k+1;
      if max < num(k) then
        max := num(k)
    until k = n
   end;
procedure printmax;
 begin
   writeln('maximum is: ', max)
 end;
begin
 readdata;
 findmax;
 printmax
end.
```

The first program is written in Prolog and the second one in Pascal.
Obviously these are only two among the possible solutions to the given problem. It is possible, for example, to implement an interactive algorithm in Prolog and also to write a recursive one in Pascal [1].
These two examples have been chosen because they reflect in some way two different styles of programming.
A Prolog program is essentially a set of logical definitions of relations. An execution of the program is a use of these definitions to compute instances of the relations. In the preceding example the program is composed of four rules expressed as sentences of predicate logic. [H|T] is a notation for the unique data structure: the list, where H indicates the head of the list and T the queue of the list.
Each rule is a true statement about the "max" relation. We can use these rules both for finding or for checking the greater of a set of numbers:
?- max([4,3,7,6], M)
?- max([4,3,7,6], 9)

A Prolog program has two different interpretations: a procedural one and a declarative one.
Following the procedural intepretation, a rule of the form A :- B, C. can be interpreted as a procedure which reduces problems of the form A to subproblems of the form B and C.
The declarative interpretation is based on the assumption that a program can be read as a theory represented in logic where ":-" means "is implied by" and "," means "and".
The two interpretations correspond to two different programming methods: "logic programming" and "algorithmic programming" [2].
Coming back to the problem of finding the maximum of a set of numbers, in logic programming style we can see "maximum" as a relation between a list of integers and a number. So a logic programming solution for maximum can be: there are three cases (1) the list is empty, (2) the list contains a single element and (3) the list contains a single element and another list. In the third case we use induction on the list.
This solution corresponds to the clauses given at the beginning of the chapter.
Due to problems related also with efficiency Prolog does not realize entirely the logic programming ideal [3] and has a number of extralogic features (such as cut to affect backtracking).
In the varions experiences about the introduction and the use of Prolog in education which have been carried out in recent years, see, for example: [2], [4], [5], [6], the declarative style was privileged: the language is used mainly to represent facts and relations between facts in given subjects. The extralogic features are used as rarely as possible, sometimes avoiding the implementation problems by means of predefined procedures which are given to students.

In the Pascal version of the "maximum" program we can see a different style of programming: we have to choose a data structure to represent data, we have to declare
the type of the variables involved, following a structured programming approach, we divide the given problem into subproblems whose solution corresponds to a number of procedures. To solve each subproblem the key role is played by assignment statements.
One of the ways which is usually followed to introduce computer science elements, and programming in particular, to secondary school students, is to insert it into the teaching of a traditional school subject (often mathematics) as a means for problem solving [7]. Knowledge is presented procedurally and the emphasis is on building procedures for performing operations which may themselves be defined in terms of other procedures. This is consistent with conventional, procedural programming, where a program consists of a set of instructions to the computer as how to perform a particular operation which may be structured in terms of procedures.
At least, at the beginning, the kind of problems and applications presented and discussed are usually numerical or similarly well defined.
The alternative declarative tradition of knowledge representation is based on clear descriptions of information often in terms of relationships between entities.
Following this approach it is possible to present students with types of problems which are different from the previous ones and to focus the attention more on the nature of the problem area, rather than on the mode of solution of a problem.

3. DISCUSSION

It is my opinion that for educational aims there does not exist a programming language which is better than another. It depends on the kind of applications involved and on the pedagogical intentions which are to be pursued.
From the point of view of the applications Prolog is useful for database construction and manipulation, and for knowledge classification and retrieval.
In the various experiences regarding its introduction and use at secondary school level, applications have been carried out in subjects which are usually influenced less by new technologies and methods, such as: humanities, history, geography, and moreover chemistry, logic, etc. For example it is used to organize historical facts or geographical data, to classify animals or given materials according to their properties, to systemize grammar rules, etc [8], [9], [10], [11].
In some cases the aim is not primarily to teach Prolog but to exploit the logical nature of the language to promote abstraction and classification capabilities, forcing students to reflect with more precision on how to represent their knowledge on a given subject.

As observed before, procedural languages such as Pascal or Basic, are usually introduced to solve numerical problems.
Applications have been carried out, for example, in the field of Mathematics or Physics, to enhance the learning of given topics, so as to implement algorithms on integer numbers, to obtain the approximated solutions of an equation, etc., and to give an operational view in the study of the subject matters [12].

In this respect it is possible to notice that with a declarative approach the kind of problems which are faced at the beginning usually require knowledge representation and modelling capabilities which are less emphasized in the first introduction of algorithmic languages, were modelling problems appear later in the learning process.
On the other hand, with declarative languages such as Prolog or Micro-Prolog, a version of Prolog which is widely used for educational applications [13], it is quite easy to begin. At first sight they look easy to learn and to use, since, using only few constructions, they give beginners the possibility to produce interesting behaviour from the machine, thereby capturing the interest of the learner from the outset. It is also possible to query a program before it is completed and to change or increase it.
Moreover languages such as Prolog, seem to meet some of the conditions that are indicated, for example from Du Boulay [14], to be necessary for the successful learning of a computer language: it is interactive, it has debugging facilities such as tracing facilities, it is based on logic rather than on any particular computer architecture, it has a simple sintax.

With procedural languages a learner is forced to a strong method and he need a great amount of rigour to respect the syntax rules.
Pascal, for example, despite its many good educational qualities, has a number of drawbacks as a first programming language. The main difficulties are the large amounts of detail which have to be mastered to make even a simple program work, the lack of debugging facilities so that a student has to laboriously insert and later remove "write" statements to produce a trace, and the fact that the language is compiled rather than interpreted which makes meaningful error reporting more difficult [14].

On the other hand, despite their apparently simplicity logic programming languages, such as Prolog, after a first easy approach, give a number of problems when it becomes necessary to gain a deeper knowledge of them.
Studies have been carried out on students problems and misconceptions. These are both conceptual problems, related to beginners' mental model of how a Prolog system operates and problems related to proper language characteristics (eg. backtracking, list
representation, recursion, etc.) [15], [16].
First of all it can be noted that for declarative languages programming techniques are still under development and especially techniques that make up the art of logic programming are still scarce. This makes their teaching difficult.
In fact, although "logic programming" is the favourite method for most expert Prolog programmers, it is not the only way to use Prolog and it is not often the preferred method of beginners. For example a study of Van Someren points out that for the class of problems he considered, the algorithmic approach is followed by beginners (both by those who had a previous experience with Pascal and also by those who were studing Prolog as a first language) [1].
For procedural programming languages, methods, such as structured programming, have been developed that could be assumed as a guide in problem solving. Moreover there exists a set of techniques (i.e. selection, iteration. etc.) that are relatively indipendent of implementation language. This produces transfer from one programming language to another.
These considerations are particularly important when the use of a programming language is seen not only as a support for the teaching of particular topics of a given subject, but also as a tool to introduce students to basic computer science ideas and methods.
In this respect it is important, for example, to understand the execution of a program, to develop a model of the programming language which is used.
For Prolog, after the first simple examples, it is not always easy to give an appropriate explanation of how one has arrived at an answer to a question. It is necessary to explain such features as backtracking, unification, recursion. Different ways could be chosen to realize this goal. H. Pain and A. Bundy in [17] point out the importance of considering the model that is to be used in teaching and to select the most important features of this model for presentation. In particular they discuss the merits of different ways of presenting Prolog search spaces, execution behaviour and answer justification, according to four different models (AND, OR trees, OR trees, Arrow diagrams and Byrd boxes). On the other hand search strategies and backtracking are complex issues which, though important to students of artificial intelligence languages, have less relevance to the problem areas described in the programs.
With traditional programming an interesting way to present students with a model of execution is to introduce them to the programming of pocket calculators which support a low level language such as Hewlet Packard calculators, like HP-11 or HP-32S, etc. [18], [19].
These kind of relatively limited instruments, which support a simple algorithmic language, makes it easier for students to obtain a thorough understanding of the computer system which is more difficult to obtain when more advanced ones are used. Moreover the use of "poor" languages give the opportunity to introduce beginners to a more indepth comprehension to fundamental concepts like: storage manipulation and conditional tests construction.

Experiences, which have been carried out following both the declarative and the procedural approach, have pointed out that the introduction of a programming language to secondary school students is a task which, though interesting and fruitful under many respect, involved a number of problems and difficulties.
For this reason it is often useful to have programming environments or front-end programs to make the first approach easier.
For Pascal, simplified environments have been realized to introduce students to the fundamental structures of the language (sequence, decision, loop) without requiring the use of data. See for example [20], [21].
For Prolog a variety of front-end programs are available offering particular styles of interaction, such as a sintax without key-words or parentheses, a window-based system, an adventures game framework, a database investigation system or interactive expert system shells [22], [23], [24].
The use of Prolog to realize expert system shells is of particular interest. In fact it is straightforward, for example, for teachers to write their own educational programs within a particular shell.

To conclude this discussion I point out that, if the objective of the introduction of a programming language to secondary school students is to give an introduction to basic computer science concepts and methods with an eye to its future developments, it is worthwhile to give students some ideas of both the procedural and the declarative approach. Experiences have pointed out that, at the age under discussion, both of them are possible and fruitful.
Later on in the learning process it will be possible to go deeper in the specific related issues, and also to compare them operatively, for example proposing students exercises to emulate the execution of a Prolog program with a more conventional language [25].

4. CONCLUSION

The focus of the paper is on the introduction of programming languages for educational aims in the first years of secondary school.
Two different kind of programming languages have been considered: procedural (esemplified through Pascal) and declarative (esemplified through Prolog).
The use of a language instead of another depends on the applications and on the pedagogical intentions which are to be pursued.
From a teaching point of view, Prolog allows to turn to such educational objectives as knowledge representation in a given domain, logic approach to problem solving, data-base construction, manipulation and query. These features enable applications in curricular areas, like the humanities, which are usually less influenced by new technologies. Moreover it seems now of increasing importance to give students an idea of fifth generation computers, logic programming and expert systems.
On the other hand, Prolog, after a first easy approach, gives a

number of problems, which are both conceptual and technical, when it is necessary to get a deeper knowledge of it.

Experiences regarding the use of procedural languages in education are, at present, more widespread. These languages are suited to develop numerical applications or conventional programming. They support a model of execution which seem easy to understand and there are also a number of languages, from those of pocket calculators to Pascal, for which transfer is possible.

REFERENCES

1. Van Someren, M. Beginners Problems in Learning Prolog, Memo n. 54, Dept. of Social Science, Informatics and Dept. of Experimental Psychology, University of Amsterdam, 1984.

2. Kowalski, R. Logic Programming as a Computer Language for Children, in New Horizon in Educational Computing, Yazdani, M. (ed), Ellis Horwood, Chichester, 1984, pp. 121-144.

3. Kowalski, R. Logic Programming, Information Processing 83, North-Holland Elsevier, 1983, pp.133-145.

4. Ennals, R. Teaching Logic as a Computer Language in Schools, in New Horizon in Educational Computing, Yazdani, M. (ed), Ellis Horwood, Chichester, 1984, pp. 164-189.

5. Maler, O., Scherz, Z., Shapiro, E. A New Approach for Introducing Prolog to Naive Users, Proceedings of the Third International Conference on Logic Programming, Lectures Notes in Computer Science N.225, Springer Verlag, 1986, pp. 544-551.

6. Bottino, R.M. Prolog Experiences in Classrooms: Possibilities and Problems - The 'PICO' Project (Prolog Information on Computers), in Prolog, Children and Students, Nichol, J., Briggs, J., Dean, J. (eds), Kogan Page, London, 1988, pp. 84-90.

7. Bottino, R.M., Forcheri, P., Molfino, M.T. Computer Science Literacy in Secondary School, Proceedings of 37 CIEAEM Conference, Leiden, 1985, pp. 283-288.

8. Ennals, R. Artificial Intelligence, Applications to Logical Reasoning and Historical Research, Ellis Horwood, Chichester, 1895.

9. Nichol, J., Briggs, J., Dean, J. (eds) Prolog, Children and Students, Kogan Page, London, 1988.

10. Centre Regional de Formation a l'Informatique, Micro-Prolog, IREM d'Aix-Marseille, Faculté des Sciences de Luminy, N.1, 1984.

11. Bottino, R.M., Molfino, M.T. Prolog as a Programming Environment for Children: some Examples, Newsletter of the Association for French Language Studies, Newcastle University, 1894, pp. 31-36.

12. Bottino, R.M., Forcheri, P., Furinghetti, F., Molfino, M.T. An Attempt of Integrating Mathematics and Computer Science, Proceedings of 38 CIEAEM Conference, Southampton, 1986, pp. 115-120.

13. Clark, K.L., Mc Cabe, F.G. Micro-Prolog: programming in Logic, Prentice-Hall, 1984.

14. Du Boulay, B. Some Difficulties of Learning to Program, Journal of Educational Computing Research, Vol. 2 (1), Baywood, 1986, pp. 57-73.

15. Taylor, J., Du Boulay, B. Why Novices May Find Programming in Prolog Hard, in The Child and the Computer: Issues for Developmental Psychology, Rutkowska, J., Crook, C. (eds), Wiley, 1986.

16. Van Someren, M. Misconceptios of Beginning Prolog Programmers, Memo N.30 of the Research Project 'The Acquisition of Expertise', Dept. of Experimental Psycology, University of Amsterdam, 1984.

17. Pain, H., Bundy, A. What Stories Should We Tell Novice Prolog Programmers?, Technical Report, University of Edinburgh, 1985.

18. Bottino, R.M., Forcheri, P., Molfino, M.T. A Work Plan to Test the Educational Opportunities for Using Small Computers in Secondary Schools, Aspects of Educational Technology Vol. XVI, Trott, A., Strongmann, H., Giddins, L. (eds), Kogan Page, London, 1983, pp. 117-121.

19. Bottino, R.M. A Course for Secondary School Teachers Concerning the Use of Programmable Pocket Calculators, Computers & Education, Vol.7, N.2, Pergamon Press, 1983, pp. 125-130.

20. Pattis, R. Programmare con Karel il Robot, Mondadori, Milano, 1984.

21. C.N.R., Corso di Introduzione all'Informatica, SEI, Torino, 1985.

22. Nichol, J., Briggs, J., Dean J., Nichol, R., O' Connell, K., Raffan, J., Wild, M. Prolog Tools in Education, in Prolog, Children and Students, Nichol, J., Briggs, J., Dean, J. (eds), Kogan Page, London, 1988, pp.144-168.

23. Scherz, Z., Weinberg, B., Pontch, M., Goldberg, D., E.E.S.S.- Educational Expert System Schell, Pegboard, Vol. 3/1, Exeter, 1988, pp. 100-103.

24. Briggs, J., Brough, D. Using Prolog in the Design of Knowledge Based tools for Education, Proceedings of PEG 88 Conference, Copenhagen, 1988.

25. Lambert, H., Pascal's Emulation of a Prolog Program, papers of the ninethenth ACM-SIGCSE technical symposium on Computer Science Education, Atlanta, Georgia, 1988, pp. 254-258.

USING CONCEPTUAL INFORMATION TO HELP STUDENTS OBTAIN EQUATIONS FOR ALGEBRA WORD PROBLEMS

M.D. LeBlanc, S. Weber Russell
Dept. of Computer Science
University of New Hampshire
Durham, NH, USA

Solving algebra word problems is considerably more complex than performing algebraic manipulation. Many students would have few problems algebraically solving an equation for x, if they could only arrive at an equation. A system which provides interactive graphics tools to help students obtain equations for algebraic word problems has previously been implemented. This paper describes the incorporation of a tutoring strategy which refers to conceptual information implicit in the word problem statement. The focus on conceptual information helps students isolate conceptual elements in the word problem as a means of bridging the transition between linguistic and algebraic representations. In the operation of the system, a conceptual analyzer first passes sentences of word problems into conceptual dependency representations. Reference to elements of these representations then allows the tutor to adjust the focus of the tutorial dialogue. For students requiring detailed remediation, a sentence-level focus addresses the potential difficulties of finding what is given in the problem and, more importantly, what is not. Higher levels of tutorial focus address the difficulties students have with recognizing the critical relationships between conceptual elements in the problem.

UTILISATION DE DONNÉES CONCEPTUELLES POUR AIDER LES ÉTUDIANTS À ÉTABLIR DES ÉQUATIONS À PARTIR D'ÉNONCÉS DE PROBLÈMES D'ALGÈBRE

Il est beaucoup plus difficile de résoudre des énoncés de problèmes d'algèbre que d'effectuer une manipulation algébrique. Un grand nombre d'étudiants éprouveraient beaucoup moins de difficultés à trouver la valeur de x dans une équation algébrique s'ils pouvaient convertir le problème en équation. Un système qui permet, grâce à des outils graphiques interactifs, de convertir en équation des énoncés de problèmes d'algèbre a été implanté antérieurement. Le présent document traite de l'introduction d'une stratégie d'apprentissage qui met l'accent sur l'information conceptuelle présente dans l'énoncé du problème. La focalisation sur l'information conceptuelle aide les étudiants à cerner les éléments conceptuels d'un problème et constitue une étape transitoire entre les représentations linguistiques et algébriques. Un analyseur conceptuel effectue d'abord une analyse des phrases du problème et les transforme en représentations en dépendance conceptuelle. La mention des éléments de ces représentations permet alors au système d'adapter la procédure d'initiation. Pour les étudiants nécessitant un soutien plus important, une fonction d'aide niveau phrase s'attaque aux difficultés potentielles qu'ils éprouvent à découvrir ce qui est donné dans le problème et, chose plus importante, ce qui ne l'est pas. Des fonctions plus évoluées s'attaquent aux difficultés qu'éprouvent les étudiants à reconnaître les liens importants entre les éléments conceptuels du problème.

USING CONCEPTUAL INFORMATION TO HELP STUDENTS OBTAIN EQUATIONS FOR ALGEBRA WORD PROBLEMS

Mark D. LeBlanc
Sylvia Weber Russell

Department of Computer Science
University of New Hampshire
Durham, NH 03824

1. Introduction

This paper describes the next step in the continued effort to build an intelligent tutor for students solving word problems. In a sense, this work represents but a small component in the overall scheme of building a machine tutor; yet, it has refocused our attention on some of the fundamental difficulties that students face when solving word problems. In many cases, it is not the mathematics that present the difficulty but rather the words which represent the problem. Since all word problem solutions start with words and eventually arrive at algebraic and numerical answers, the tutor must have access to the linguistic representation. Our focus on conceptual information supports our goal of explaining the transition from the language of words to the language of algebra.

The emphasis on this transition is novel in its attempt to help students who do not know where to begin. While many non-tutoring systems "read" and completely solve word problems (Bobrow, 1968; Novak, 1977; Bregar, *et al.*, 1986), our effort is an attempt to teach the strategies which isolate and organize the information needed for an equation. If we want to tutor students who are solving word problems, we can not overlook the *process* of reading, rereading, and isolating important information.

A focus on the transition *away from* the linguistic representation requires a fundamental concern with the tutor's ability to provide hints and generate explanations about this process. This means, first, that the tutor must have access to the mapping between a lexical item or items and its corresponding algebraic equivalent. For example, *"at 60 mph"* maps into (speed (value 60) (unit mph)) and *"How far"* maps into (distance (value ?) (unit ?)). Second, after the tutor *reads* (hereafter analogous with *parses*) the word problem, a set of conceptual items must be available to the tutor in order to provide a means of initiating questions to the student. Last and most ambitious, the tutor should be able to explain *how* it found the conceptual items as it parsed the word problem. As currently implemented, the prototype parser maps sentences of word problems into an intermediate conceptual representation which provides the first two requirements: the lexical-to-conceptual mapping and a set of conceptual elements for generating questions. The third requirement, that of generating explanations of the parsing process, is done only to a rough approximation, i.e., the explanations are not in a form suitable for student reading.

The paper proceeds as follows. A sample student-tutor session highlights the potential use of the conceptual information. Section three outlines the idea of conceptual analysis and our prototype parser. We close with a discussion of where this work has led us and whether it was worth the trip.

2. An Example Student-Tutor Session

Given a word problem to work on, the tutor, as currently prototyped, encourages the student to reduce the problem to a number of subproblems of a more manageable size. Working in this problem reduction mode, students are introduced to tasks commonly done by experienced problem solvers, which include *Finding a Relationship*, *Draw a Picture*, and *Organize the Data*. Interactive tools are then provided to facilitate the completion of specific tasks (LeBlanc, 1988).

In the hypothetical student-tutor protocol shown below, the student has previously been shown the word problem,

> **A train leaves a station at 60 mph.**
> **How far does the train go in 3 hours?**

and has decided to organize the data into the known and unknown values. The script begins at the point where the tutor (T) introduces the task, *Organize the Data*, to the student (S).

T: To help you establish an equation, it is a good strategy to sort out what you know and what you don't know.

[Because we feel that conceptual elements benefit mainly the discouraged problem solver, the tutor in this example interacts in remedial mode. Of course, not all students will

require such detailed tutoring. However, the intended audience is not those who already know how to solve these types of problems but rather those who sigh and exclaim "I never could do math.")

T: Let's try to find something in the first sentence which might help us.

A train leaves a station at 60 mph.

(*Having parsed the sentence into conceptual items, the tutor has a set of potential questions.*)

T: What is moving?
S: the train

(*The same parser which interprets the word problem is used to interpret student input.*)

T: Ok, where does the train start?
S: station

T: Good. Do we know where the train is going?
S: no

T: Correct, the sentence doesn't tell us. How fast is the train going?
S: 60

T: 60 what?
S: ?
T: What are the units of speed? 60 ____ ?
S: mph

T: Now let's look at the second sentence.

How far does the train go in 3 hours?

T: Is the question asking for a distance, a rate, or a time?
S: distance
T: What are the two words which tell you that?

(*For each of the conceptual items in the problem, the parser remembers which word(s) contributed to the establishment of that conceptual item.*)

S: how far

T: How long does the the train travel?
S: 3 hours
T: Ok, we now know

	rate = 60 mph
train:	time = 3 hours
	how far = distance = ?

(*Actually, the data would be presented in a table format. As the student identifies information, that entry is added to the table. At this point the tutor would successfully return from the task, Organize the Data.*)

3. The Conceptual Analyzer

The conceptual analysis of sentences is the process of mapping a natural language input such as English into an internal *meaning* representation. At any point during the conceptual parse of a sentence, the parser has a meaning representation of what it has *read* thus far along with a set of expectations of what is to come. The next two sections of the paper provide the background for conceptual analysis and a sample sentence analysis.

3.1. Background of Conceptual Analysis

Conceptual analysis as a parsing method has been used in story understanding (Schank and Abelson, 1977; Riesbeck and Schank, 1979). Short stories were parsed into Schank's conceptual dependency (CD) representation and ultimately represented in scripts, where a script was a collection of expected events which typically occurred in a specific situation. The combination of CDs and script representations provided a basis for story paraphrasing and question-answering (Lehnert, 1977). The initial conceptual analyzer (Riesbeck, 1975) was later extended by Birnbaum and Selfridge (1981) to handle word-sense disambiguation and to include Gershman's (1977) analysis of noun-groups (e.g. *the large blue passenger train*).

3.2. Parsing Word Problems

The prototype parser, an adaptation of the Conceptual Analyzer of Bimbaum and Selfridge (1981), maps sentences of a word problem into CD-representations, using simplified versions of the noun-group and disambiguation features mentioned above. Each word is associated with a set of situation-action rules called requests. When a particular word is read, its requests become active. Requests may look for a specific word or phrase in the input or they may look for some more general conceptual item such as a physical object that can move. The requests highlight the important feature which makes conceptual analyzers, in general, *expectation-based*. At any one point during the parse of a sentence, the state of affairs consists of 1) a set of conceptual elements from the words already read (stored in the parsers short-term memory) and 2) a set of expectations about what conceptual items might come later on. What makes this useful to a word problem tutor is the fact that this is, in a large part, how we as humans read and interpret sentences. For example, if I say to you, *"A train leaves ..."*, you can tell me that you know a train begins moving and that it will start *from* somewhere and will *probably* travel *to* somewhere else *at* some rate of speed, etc. This is exactly what the parser would tell you if the analysis were stopped after the word *"leaves"*.

A brief handtrace will help to explain how the parser maps the first sentence of the word problem in Section two, **A train leaves a station at 60 mph,** into a set of conceptual elements. It is worth emphasizing that it is not only the output of conceptual information which is needed by the tutor. Access to the *process* of expecting, finding, and joining conceptual items is essential if the tutor is ever

expected to provide a reason of how *it* would read the sentence.

The word A is read and retrieved from a dictionary of words. A request attached to this word is generated which tests whether some *object* will appear later on in the sentence. If that situation is met then the associated action will add *(indefinite reference)* to that *object*. The word train is read and an action associated with the word train stores a conceptual item in short-term memory (STM) with the following characteristics: a physical object which can move but is not alive. The expectation generated by the word A is now satisfied. An *object* has been found and the conceptual item for train in the STM is marked as an object not previously seen.

The word leaves is read. In the dictionary, the verb leaves can be conceptualized as one of two possible primitive acts: PTRANS, a physical transfer of location, or ATRANS, an abstract transfer of possession, e.g. *John leaves a book on the table*. In this sentence, the parser can disambiguate between the two potential acts since inanimate physical objects (i.e., the train) do not transfer possession. The selected PTRANS-sense of leaves stores in the STM that an *actor* (someone or something) changed location *from* some starting position *to* some unknown position *at* some rate of speed, etc. Requests are generated which expect to find the actor which is moving, the starting and ending locations, a speed, the length of the move, etc. The first of these requests is immediately satisfied since the conceptual element for train already in the STM satisfies the actor requirement. The other requests serve as expectations of what might come later in the sentence.

The phrase a station is handled in much the same manner as A train. A conceptual element which serves as a *location* is stored in the STM. This satisfies the outstanding expectation that is looking for a starting point for the object that moved. The current status of the STM now reveals that a physical object moved from this location.

The phrase at 60 mph, while handled word-by-word, eventually collapses to a *speed-unit* concept with a value of 60 and a unit of miles per hour. Since the parser is expecting such a conceptual item to appear, this item fills the speed slot associated with the PTRANS act. The complete output of the parser for this sentence is shown below.

SHORT-TERM MEMORY after reading the sentence
A train leaves a station at 60 mph.

ACT: ((PTRANS (phrase (leaves)))

ACTOR: ((class physical_object) (reference indefinite)
(features (-fixed -animate)) (phrase (a train)))

OBJECT: {same as actor}

FROM: ((class locale) (reference indefinite)
(features (+fixed -animate)) (phrase (a station)))

TO: (?)

SPEED: ((value 60) (unit mph)
(phrase (at 60 mph)))

Conceptually we have the interpretation that *a physical object which can move but is not alive has transferred itself from a location which can not move and is not alive to some unknown location at a speed of 60 mph units.*

4. Discussion

The decision to turn our attention to the parsing problem was not done without careful consideration. After all, parsing natural language still remains very open-ended, even within restricted domains. Our parser, while robust enough to differentiate between the meaning of the sentences *A man leaves a station* and *A man leaves a tip* does not by itself provide the "reading potential" that a word problem tutor requires. So the question arises, are we on a path worth pursuing? We think the answer is yes and we close with some considerations which influenced our decision.

First and foremost, a word problem tutor must have a capability to explain the transition *from* the words to some intermediate representation, whatever that might be. The difficulty with word problems is of course not solely a reading problem (Barlow, 1964; Knifong and Holtan, 1976). However, the word problem solver must start with a linguistic representation. Successful problem solvers represent word problems in a number of alternate representations, such as pictures, tables, or even incomplete equations. Each of these is a potential strategy for attempting to arrive at an equation. Pictures and tables have already been incorporated into the current tutor. In addition, another tutorial component to address the transition from the isolated conceptual information to the equations is foreseen. This component will help students to see events and situations described by CD representations in a form which resembles more closely the desired symbolic equations. This form corresponds to a representation, more general and abstract than that of CD, which is based on critical relationships between conceptual elements in the problem. If we hope to encourage students to try these strategies, then the tutor must be capable of suggesting how and where to find the pieces needed for such alternate representations.

A second point is that conceptual dependency theory offers the tutor a representation which can support operations essential to the tutoring process. In the above presentation we have introduced a parser which can analyze a word problem and produce as output the conceptual meaning units of CD theory. This parsing process approximates human comprehension in that the parser does not spend its time building syntax-based sentence diagrams but rather conceptual objects which can facilitate the words-to-equation transition. Once scripts are added to the capabilities of the parser, the system can be used to answer student's questions about the problem. The output of our

parser supplies some of the basic knowledge a tutor needs to successfully understand, make inferences, and reply to student queries.

Finally, the parsing prototype has refocused our attention on the problem students have with variables. Frustrated problem solvers spend a good portion of their time trying to figure out what they don't know and we as tutors seem to spend a good deal of our time providing hints which might lead to the discovery of the problem's unknown. Granted, parsing alone will not provide our intelligent tutors with a process which guides students to this discovery. But it does find the pieces. Knowing what was found and how they were found are the prerequisites for explaining the process.

Bibliography

Barlow, I.H. (1964). Reading and computation ability as determinants of problem solving. *Arithmetic Teacher*, Vol. 11, pp. 18-22.

Birnbaum, L. and Selfridge, M. (1981). Conceptual analysis of natural language, in R. Schank and C. Riesbeck (eds.), *Inside Computer Understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Bobrow, D.G. (1968). Natural language input for a computer problem-solving system, in M. Minsky (ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press.

Bregar, W.S., Farley, A.M., and Bayley, G. (1986). Knowledge sources for an intelligent algebra tutor. *Computational Intelligence*, Vol. 2, pp. 117-129.

Gershman, A.V. (1977). Conceptual analysis of noun groups in English. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. pp. 132-138.

Knifong, J.D. and Holtan, B. (1976). An analysis of children's written solutions to word problems. *Journal for Research in Mathematics Education*, Vol. 7, pp. 106-112.

LeBlanc, M.D. (1988). Instructional Tools for Algebra Word Problems. *Proceedings of Intelligent Tutoring Systems (ITS-88)*. Montreal, Canada. pp. 238-242.

Lehnert, W. (1977). *The Process of Question-Answering*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Novak, G.S. (1977). Representation of knowledge in a program for solving physics problems. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. pp. 286-291.

Riesbeck, C. (1975). Conceptual analysis, in R.C. Schank (ed.), *Conceptual Information Processing*. Amsterdam, Netherlands: North Holland.

Riesbeck, C. and Schank, R.C. (1979). Comprehension by computer: Expectation-based analysis of sentences in context, in W.J.M. Levelt and G.B. Flores d'Arcais (eds.), *Studies in the Perception of Language*. Chichester, England: John Wiley and Sons Ltd.

Schank, R. and Abelson, R.P. (1977). *Scripts Plans Goals and Understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates.

AN INTERACTIVE ENVIRONMENT TO TEACH DEDUCTIVE LOGIC

E.H. Haeusler
Dpto de Computaçao
Universidade Federal Fluminense
Niteroi, Brazil

A great deal of the mathematical reasoning is based on the logical reasoning. So, an environment to teach operationally how to prove logical theorems will be a great aid to both teachings. The Natural Deduction (N.D.) system (Prawitz 65) reproduces the way a mathematician proves theorems. We have a theorem prover based on game-theoretic semantics (Haeusler & Pequeno 88) which, in fact, works by implementing the N.D. rules in a game-like way.

The environment proposed in this work is built by replacing one player by the user. Thus a theorem proving proceeds as a user-computer dialogue. The dialogue has a scheme, inherited from the N.D. rules, that drives the user choices according to the formula format. The dialogue itself shows to the user the consequences of its choices, giving him the operational meaning of the N.D. rules. In other words, it interactively teaches how to prove theorems.

UN MILIEU INTERACTIF POUR ENSEIGNER LA LOGIQUE DÉDUCTIVE

Une bonne part du raisonnement mathématique est fondé sur le raisonnement logique. Un cadre permettant d'enseigner de manière opérationnelle à démontrer les théorèmes logiques sera donc d'une grande utilité aux deux méthodes d'enseignement. Le système (Prawitz 65) de déduction naturelle (N.D.) reproduit la façon dont un mathématicien démontre les théorèmes. Nous avons un système de démonstration des théorèmes basé sur la sémantique du jeu et de la théorie (Haeusler & Pequeno 88) qui, en fait, fonctionne en mettant en application les règles de D.N. a la façon d'un jeu.

Le cadre proposé est établi en remplaçant un joueur par l'utilisateur. Le processus de démonstration d'un théorème se déroule donc comme un dialogue utilisateur-ordinateur. Ce dialogue suit un plan, tracé par les règles de D.N., qui dicte les choix de l'utilisateur selon la structure de la formule. Le dialogue lui-même montre à l'utilisateur les conséquences de ses choix en lui donnant la signification opérationnelle des règles de D.N. Autrement dit, il enseigne de manière interactive à demontrer les théorèmes.

AN INTERACTIVE ENVIRONMENT TO TEACH DEDUCTIVE LOGIC

Edward Hermann Haeusler*

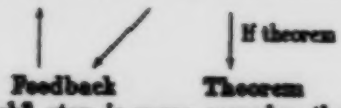
Univ. Fed. Fluminense - Brasil

I. Introduction

This work presents an interactive environment to teach deductive logic. The environment proposed here is directed to teaching logic with mathematical purposes. So, in order to justify its mentioned use we assume two facts. However we do not discuss such assumptions deeply, since it is out of this work's scope.

- A great deal of the mathematical reasoning is based on the logical reasoning.

For, the common mathematical work follows roughly the below diagram :



The "Feedback" step is more complex than we could think as showed Lakatos in [3]. It seems to be more related to the creative part of the mathematical work. We can say that all the non-creative or even mechanic reasoning is very concerned to acts of proving.

Even if the reader does not agree to the "meaning as use" semantic due to Wittgenstein, she must agree, at least partially, that the teaching of logic with mathematical purposes must not involve questions that are either ontological or deeper epistemological, very usual in logic, since the goal is learning the mathematical practice.

- The teaching of logic, at least with mathematical purposes, must be based on the "meaning as use" semantic.

Due to the second assumption we only worry about deductive logic, and apply the "meaning as use" to it. The system of Natural Deduction [4] takes into account the "meaning as use" semantic in its definition. Thus, our interactive environemnt is indeed a Natural Deduction theorem prover implemented in a game-like way adapted to play a theorem candidate with the user, explaing its decisions at each move in terms of logic.

*Partially supported by CAPES and SID informatica project ESTRA

II. The Natural Deduction System.

The Natural Deduction (N.D.) deductive system [4] was created intending, among other goals, to reproduce the way a mathematician proves theorems. Its main feature is to have groups of inference rules for each logical constant, such that each group gives in a certain way the operational meaning of the logical constant associated. Such operational meaning is attained by giving conditions in which we can introduce the logical constant in the discurse and giving conditions from which we can deduce things from a logical statements which has the logical constant as main. For example :

- *The logical implication.* The usual way a mathematician uses to prove an expression of the form $A \rightarrow B$ is *to assume A as hypothesis and prove B*. This is the condition to introduce the $\rightarrow$. The reader must note that although the proof of B is depending on A, the proof of $A \rightarrow B$ does not depend on A, since this dependence is still implicit in the meaning of $\rightarrow$. Similary, the condition to eliminate the $\rightarrow$ from $A \rightarrow B$ is $A \rightarrow B$ itself and A.

$$\frac{\begin{array}{c}[A]\\ \Pi \\ B\end{array}}{A \rightarrow B} \qquad \frac{A \; A \rightarrow B}{B}$$

where Π is any proof of B from A and the $[.]$ indicates the discharge of the assumptions as explained above. The above inferences are $\rightarrow$-introduction and $\rightarrow$-elimination rule respectively.

Below we give some N.D. rules :

$$\frac{\begin{array}{c}\Pi \\ A(a)\end{array}}{\forall x A(x)} \qquad \frac{\forall x A(x)}{A(t)}$$

Likely, the right rule is called $\forall$-introduction. It has a condition to be sound, that stated in english is : *no assumption present in the proof Π of $A(a)$ can have the parameter a.* This mirrors perfectly the mathematical practice in proving universals, that is, in order to proof that a fact

is universal we prove it to an arbitrary object. This arbitrary object is represented here by "a".

For the $\wedge$-introduction we have that from A and from B we can deduce $A \wedge B$. The $\wedge$-eliminations are the dual ($A \wedge B \vdash A$, $A \wedge B \vdash B$) rules.

III. The Theorem Prover.

We have a complete and sound (w.r.t. classical logic) automatic theorem proving based on game-theoretic semantics which produces N.D. proofs [1]. Indeed, it can be viewed as a N.D. theorem prover. It works by implementing the N.D. rules as well derived rules for the logical negation in a game-like way. This game has two players : one (myself) tries to prove the formula, while the other (nature) tries to prove its negation.

The idea of the game-theoretic semantics, due to Hintikka, is to assign the truth value of a formula depending on the result of a zero-sum, two opponents game. A game is defined for a given formula and its interpretation. If myself wins the game, the formula is assigned true for the given interpretation. If the nature wins, it is assigned false. In a classical setting, draw is not allowed for the excluded-middle principle.

Thus the game is performed by recursion on the formula's degree. If the formula is atomic, the interpretation is used to evaluate the truthfulness of the formula and the winner is myself if it is truth. The other steps are performed by choices of the players on the subformulas. The nature choices when the logical constant are either the $\wedge$ or the $\vee$ and clearly the negation causes the swapping of the players. In order to work as a truth definition the game is highly idealised. The players are assumed to be impecable. They always take the best moves. It does mean that the victory of a player can be confounded with the existence of a winning strategy for him. Note that a little of classical semantic is used in the atomic step.

The use we make of games is not exactly as above. The problem our theorem prover is concerned is not the verification of the satisfation of a formula by a given modal but the answer of the question on whether a formula (or its negation) is a theorem of a knowledge base given by axioms. Thus we are not restricted to consider just one modal but the whole class of the models of the base. We will work then with a class of games, one for each modal of the base. In order to be considered a winner one has to win all the games. The formula will be a theorem of the base if myself wins all games. Its negation will be a theorem if the nature wins all the games. It may happen that myself wins some of the games and nature others, this will be considered a draw. It means that the knowledge is incomplete : neither the formula nor its negation are theorems of the base.

The effect of playing a class of games is achieved in the method by the use of two games. One, named Agame (for Analytical game), works, as the game exposed before, by decomposing the formula in its prime components. The other game, named Bgame (for Basegame), is to be played with an atomic formula and the base. This last game finds out whether the atomic formula (or its negation) can be deduced from the base. This idea was firstly proposed by Jackson in [2].

The Agame works on the formula to decompose it into its atoms. Each atom is then proposed to the Bgame, that plays it against the base. These two games are mutually recursive : the Bgame may call back the Agame, that calls the Bgame and so on.

Considering the Agame first, let p be used to designate the ap (affirmative player or myself) or the np (negative player or nature). We will use the function "opn" to designate the opponent of a player.

1. Agame(A,p). If A is atomic, p plays the Bgame for A and the knowledge base KB. If p wins this game, it is the winner of the Agame, and A is a theorem. If p loses , opn(p) is the winner and $\neg A$ is a theorem. It may happen also the draw which means the incompleteness of the base with respect to A.

2. Agame($\neg A$,p). The opponents swap their places and the game continues as Agame(A,opn(p)).

3. Agame($A1 \rightarrow A2$,p). The game proceeds as Agame($A1$,p) , then :

   (a) If opn(p) wins then p wins

   (b) If p wins, then Agame($A2$,p) is played who wins this game is the great winner.

   (c) If a draw happened then we add $A1$ to the base (KB) and the game proceeds as Agame($A2$,p). In this case p wins iff it wins Agame($A2$,p).

   We note that the first and third items of the game for the $\rightarrow$ mirrors the $\rightarrow$-introduction rule. The relationship to the first item is evident, while to the third it is sufficient to see that it is based on the fact that from a falsehood we can deduce anything (B in our case) and thus using an $\rightarrow$ we deduce the desired implication. The second item reflects

the structure of the following derived rule : from A and $\neg B$ we can deduce $\neg(A \rightarrow B)$.

The $\wedge$ introduction rule and its derived rules for the negation ($\neg A \vdash \neg(A \wedge B)$ and $\neg B \vdash \neg(A \wedge B)$) are implemented by the following :

4. Agame($A1 \wedge A2$,p). The opponent opn(p) chooses one of the formulas (Ai) then the game proceeds as Agame(Ai,p), if opn(p) wins then it wins the game, if p wins then the game proceeds as Agame(Aj,p) ($j \neq i$) and if opn(p) wins this last game it wins the game, if p wins such game then p wins the game.

5. Agame($\forall x A(x)$,p). The game proceeds as Agame($A(x)$,p) and if p wins then p wins the whole game, otherwise the game proceeds as Agame($A(c-x)$,p) and opn(p) wins the whole game iff it wins this last game.

It is interesting to explain better the above item as an implementation of the $\forall$-introduction rule. The variable x in the game instance Agame($A(x)$,p) is not indeed a variable, since we forbid its substitution in the game sequel. Thus, if p wins this game it does mean that $A(x)$ can be proved and since x is not a result of a substitution we are in conditions to apply the $\forall$-introduction rule. The difference between the $c-x$ and the x is that the former can be replaced by a term, it is really a variable. Thus, if opn(p) wins $A(c-x)$ it does mean that $\neg A(t)$ can be proved for some term t. Thus we can prove $\exists x \neg A(x)$ and by a derived rule for the negation of the universal ($\exists x \neg A(x) \vdash \neg \forall x A(x)$) we get the desired result. Note that the substitutition is the counterpart of the opponent's choice in terms of game-theoretic semantic.

The Bgame works on the base (KB). It selects axioms from the KB that are relevant to the determination of either A or $\neg A$, it plays sucessive games until one of the players wins or the axioms are exausted. Since A is already atomic, the analysis is performed on the structure of the KB formulas. However, in order to be sound the analysis of a KB formula must be done in such way that the negations are passed along its subformulas. Thus we use a flag in the game to inform whether the formula under analysis is under the scope of a negation. Let B be a selected KB axiom.

1. If B is atomic, p wins Bgame(A,B,p,flag) if A and B are unifiable (in Robinson's sense).

   Note that in this point we can apply substitutions to the existential variables ($s-x$).

2. Bgame($A, B1 \wedge B2$,p,flag). If flag informs that $B1 \wedge B2$ is within the scope of a negation, it continues as game-$\vee$($A, B1, B2$,p,flag), else it continues as game-$\wedge$($A, B1, B2$,p,flag)

3. Bgame($A, B1 \vee B2$,p, flag). If flag informs that $B1 \vee B2$ is within the scope of a negation, it continues as game-$\wedge$($A, B1, B2$,p,flag), else it continues as game-$\vee$($A, B1, B2$,p,flag)

4. Bgame($A, \neg B$,p,flag). The game continues as Bgame(A, B,opn(p),$\neg$flag).

5. Bgame($A, \forall x B(x)$,p,flag) continues as :

   (a) Bgame($A, B(x)$,p,flag) if flag informs no negation.

   (b) Bgame($A, B(c)$,p,flag), where c is a new constant in the game, if flag informs that there is a negation.

The first item of this last item reflects the $\forall$-elimination rule, reminding here that x in this case has really the role of a variable. The second item represents a derived rule for the elimination of the $\forall$ occuring within the scope of a negation using an $\exists$-elimination rule applied to a negated formula.

game-$\vee$($A, B1, B2$,p,flag). Whoever wins Bgame($A, B1$,p,flag) or Bgame($A, B2$,p,flag) (the choice is made by p), must win Agame($\neg B2$,winner) and Agame($\neg B1$,winner) respectively in order to win the game-$\vee$.

The above item represents the rule of the disjunctive silogism ($A \wedge (\neg A \vee B) \vdash B$), here as the elimination rule for the $\vee$. This rule caused some drawbacks in the completeness of the theorem prover originally proposed in [2], thus we introduced in it a more sophisticated mechanism in order to solve it (see [1]).

game-$\wedge$($A, B1, B2$,p,flag). Whoever wins Bgame($A, B1$,p,flag) or Bgame($A, B2$,p,flag) (the choice is made by p), wins the game.

We would like to point out that with the unrestricted backtracking the theorem prover is complete with respect to classical logic. Its soundness is a consequence of the relationship with N.D.

IV. The Environment.

The environment proposed in this work is built by replacing one player (the one which tries to prove the formula) by the user. Thus a theorem proving proceeds as a user-computer dialog. The dialog has a scheme, inherited of the N.D. rules, that drives the user choices according to the formula format. The dialog it-

self shows to the user the consequences of her choices, giving her the operational meaning of the N.D. rules. In other words, it interactively teachs how to prove theorems. In order to explain the environment we show a proof done by the theorem prover searching the possible dialogs between the user and the environment.

The user tries to prove $(A \wedge B) \rightarrow (B \vee C)$ from an empty KB. The environment calls the theorem prover:

Agame$((A \wedge B) \rightarrow (B \vee C)$,ap)

Since $\rightarrow$ is the main connective in the formula, it uses the game-rule about the implication. This game-rule does not have any choice to be made and following it the antecedent must be the instance of a possible recursive call. However, the environment, now responsable by the theorem prover control, checks the KB and see its emptiness. Thus, it sends a message to the environment interface that says to the user that unless she wants to find out whether $A \wedge B$ is a falsehood she must not try to continue by this way. Let's assume that the user accepted the checking. Thus :

Agame$(A \wedge B$,ap)

Thus, following the corresponding game-rule, the theorem prover as the opponent of ap chooses some of A or B sending a message to the environment interface that explains to the user it is trying to falsify $A \wedge B$. However, any case the theorem prover calls an atomic instance of Agame with the KB empty, sending a message to the interface saying that an atomic formula cannot be a logical consequence of nothing.

Thus, following the implication's game-rule, the theorem prover sends a message to the user saying that she must try for assume the antecedent $A \wedge B$ of the implication and prove the sucedent $B \vee C$. So, it adds $A \wedge B$ to the KB and does the call :

Agame$(B \vee C$,ap)

Following the corresponding game-rule the theorem prover sends a message to the interface saying to the user that in order to prove a disjuntion she must prove one of its elements, so she must choose one of them. Let's assume that she chooses C, then the interface sends a message to the theorem prover continues with C. Thus Agame$(C$,ap) with KB $= \{A \wedge B\}$ is the next call. Thus the theorem prover see that the set of relevance of C with respect to the current KB is empty. Again, it is impossible to prove or disprove. Thus, Agame$(B$,ap) is the next call. In this case there is a set of relevance and (in general) the user is invited to choose some element of it. Thus :

Bgame$(B, A \wedge B$,ap,no-negation)

Some explanation about the $\wedge$-elimination is given to the user and she is invited to choose between A and B. Let's assume that this time she paid more attention and B was choosen. Thus, finally the theorem prover sends a message to the interface saying to the user that she proved the theorem, since "ap" won the game. And if the user wants to see the proof the sequel of applications of N.D. rules in the successful path is showed to her.

V. Conclusion.

By a analysis about the above example the reader can note that the environment follows the steps of the theorem prover explaining them to the user and asking the necessary choices. The instructional power is in the N.D. rules structure. Thus the reader can see that the environment proposed here agrees to the requirements we made for the teaching of logic, at least with mathematical purposes.

We can remark that some refinaments may be done in the environemt. For example it can have different levels of dialog depending on the user knowledge. We can also expand the environment such that it works with natural language stated utterances instead of logical language. So it will be more friendly. But we remind that a total translation to natural language maybe lead to the inverse effect.

Finally we point out that the idea could be adapted to other ways of linguistic reasoning since we had a N.D. theorem prover for the logic that reflected the desired way of reasoning.

Acknowledgments.I am very grateful to Tarcisio H. C. Pequeno for the discussions we had during the theoretical development of this work as well as for the incentive he gave me to write this article.

REFERENCES

1. Haeusler, E. H. & Pequeno, T. H. - Explanation by Game, in Proceedings of the ITS-88 international conference , 1988, Montreal Canada.

2. Jackson, Peter - Implementing a Game-Theoretic Interpretation of Logic. DAI Research Paper C263, University of Edinburgh, 1985.

3. Lakatos, Imre - Proofs and Refutations - The logic of Mathematical Discovering. Cambridge University Press, 1976.

4. Prawitz, Dag - Natural Deduction. Stockholm, 1965.

INSTRUCTIONAL PLANNING BASED ON FUZZY REASONING: AN ICAI COMPONENT

J. Liu
McGill Research Centre for Intelligent Machines
McGill University
Montreal, PQ

G.M. Boyd
Dept. of Education
Concordia University
Montreal, PQ

In this paper, we present a new approach to instructional planning in ICAI systems where sufficient knowledge about instructional decisions is characterized as being acceptably vague and incomplete. Our review shows that the inadequate planning capability of existing ICAI systems (and of conventional CAI) is largely because they make instructional decisions based on deterministic algorithms and/or deductive reasoning using explicit decision rules.

As an alternative, we explore the feasibility of inferring sub-optimal instructional tactics and operations choices from implicit and fuzzy knowledge.

The major technical result obtained from the study of a prototype instructional planning system is to show that this approach, which requires only minimum high-level abstract or vague knowledge about instruction, can still yield satisfactorily explicit instructional decisions in various situations and, therefore, should be recommended as a component for future ICAI systems design.

PLANIFICATION DE L'INSTRUCTION BASÉE SUR LE RAISONNEMENT FLOU: ÉLÉMENT D'EIAO

Ce document contient une nouvelle approche à la planification de l'instruction dans les systèmes EIAO, où une connaissance suffisante des décisions en matière d'instruction est définie comme étant raisonnablement vague et incomplète. Notre étude révèle que la capacité de planification insuffisante des systèmes EIAO (et des systèmes EAO conventionnels) actuellement utilisés est en grande partie due au fait que les décisions adoptées en matière d'instruction sont fondées sur des algorithmes déterministes ou sur un raisonnement déductif basé sur des règles de décision explicites.

Une autre façon de procéder consiste à étudier la possibilité de déduire des choix d'opérations et des tactiques pédagogiques sub-optimales à partir de connaissances implicites ou floues.

Le résultat technique le plus important de l'étude d'un prototype de système de planification de l'instruction est que cette façon de procéder, qui ne nécessite qu'un minimum de connaissances abstraites ou vagues de haut niveau concernant l'instruction, peut néanmoins permettre de dégager des décisions suffisamment explicites en matière d'instruction dans diverses situations et devrait par conséquent être recommandée à titre d'élément de la conception des systèmes EIAO futurs.

Instructional Planning Based on Fuzzy Reasoning: An ICAI Component

Jiming Liu * Gary M. Boyd ‡

* McGill Research Centre for Intelligent Machines
McGill University
Montreal, Quebec Canada
‡ Department of Education
Concordia University
Montreal Quebec Canada

Abstract

In this paper, we present a new approach to instructional planning in ICAI systems where sufficient knowledge about instructional decisions is characterized as being acceptably *vague* and *incomplete*. Our review shows that the inadequate planning capability of existing ICAI systems (and of conventional CAI systems) is largely because they make instructional decisions based on deterministic algorithms and/or deductive reasoning using explicit decision rules.

As an alternative, we explore the feasibility of *inferring* sub-optimal instructional tactics and operations from implicit and fuzzy knowledge.

The major technical result obtained from the study of a prototype instructional planning system shows that this approach, which requires only minimum high-level abstract or vague knowledge about instruction, can still yield satisfactorily explicit instructional decisions in various situations and therefore should be recommended as a useful technique for future ICAI systems design.

1. Introduction

Theoretically speaking, an *intelligent computer aided instruction* (ICAI) system would achieve the adaptability and flexibility of its instructional planning by maintaining three separate, but interrelated, components as interaction goes on [1][2][3][5]. These three major components can be outlined as follows:

- A sufficient model of the subject being taught for the system to follow the main line of the student's performance;
- A student's learning model which determines the underlying knowledge or skill deficiencies which lead to the observed responses if the student's answers are incorrect or unanticipated;
- An instructional planning component which generates appropriate instructional sequences and/or operations in such ways that they can be adapted to students' needs and lead to effective instruction, by providing tasks and feedback which are performance- sensitive.

Over the past ten years, there have been several studies focused on the construction of a student model that can obtain a rough understanding of the student's progress towards mastery and this knowledge might further be used to tailor the instructional operations to the individual needs of the student. Examples of ICAI systems which possess such a model are Brown and Burton's WEST [10]and Goldstein, Stansfield and Carr's WUMPUS [7]. There are also some studies concentrated on the development of domain expertise such as Stevens and Collins' WHY [4]and SCHOLAR [9], and Brown and Burton's SOPHIE [10].

Nevertheless, as Self has pointed out [8], few studies have been done with respect to the third ICAI component - the *instructional planning component*. The existing ICAI systems, which have included certain planning facilities, can only reason about very limited and explicit instructional strategic rules and hence cannot deal with many complex situations [19]. This is mainly due to the unavailability of detailed decision rules that can be immediately applied to guide the generation and selection of specific instructional operations and also due to the lack of some effective way of representing and reasoning about human vague instructional strategic knowledge such as general teaching heuristics.

In this study, attention has been paid directly to the development of a new practicable planning technique, one which incorporates *fuzzy reasoning based upon high-level instructional strategies* to generate specific instructional tactics. The theoretical framework for the underlying representation and reasoning technique draws on Zadeh's theories of fuzzy sets and fuzzy reasoning [12].

2. Ideally Vague Strategic Knowledge-Based Instructional Planning

Outputs from an instructional planning component should include decisions specifying the instructional operations to be used with a student under some particular conditions. But in order to support such decision-making activities, the planning component must have a built-in representation of instructional strategies and tactics. In other words, building an effective instructional planning component to a large extent consists of embedding instructional strategies which might be somehow labelled to enable appropriate selection.

However, because of the vagueness of instructional strategies [6][11], any system generating and selecting tactics and actions based on those strategies will inevitably be confronted with a very difficult problem. This is the problem of using the evaluation criteria and conditions in the vague propositions. A vague term is one referring to a quality or property that things may have in varying degrees. It lacks precise limits.

Although vague strategic knowledge does not specify instructional operations deterministically, it may prioritize a loosely bounded set of possible choices and thereby weakly specify preferable selections, organization, and scheduling of operations to be performed. But, in order to understand such vague strategic specifications, it requires some effective techniques for approximate representation and reasoning

In existing ICAI systems, the fuzziness of the knowledge base is usually ignored simply because neither predicate logic nor probability-based techniques can provide a practicable basis for dealing with it. As a consequence, what are really fuzzy facts and rules are manipulated as if they were nonfuzzy. This has led to conclusions and prescriptions whose validity is open to question. In most of the existing ICAI systems, for example, the evaluation of a student's performance is solely dependent on the analysis of his or her response (albeit possibly extended natural-language response) to a given question or to series of similar questions. Such a narrowly response-specific evaluation is not usually adequate for diagnosing the deep structure of student's understanding

3. Fuzzy Production-Rule Representation and Reasoning

To cope with uncertainty in human language, Zadeh proposed, in his classic papers [12][13], a theory of fuzzy sets. More importantly, he formulated a theory of fuzzy reasoning which can provide a basis for reasoning and deduction under vagueness. The basic idea behind the theory of fuzzy sets is that one should model directly the intuition of a dispersive prime proposition, i.e. the existence of borderline cases of set membership. According to Zadeh, the notion of "meaning" can be formalized by equating it with a fuzzy set on a universe of discourse generated by a kernel space. A kernel space **K** can be any prescribed set of objects or constructs.

Primarily, there are four types of fuzzy propositions that one can use in performing fuzzy reasoning [13]: namely:

- An unconditional, unqualified proposition: **X** is **F**;
- An unconditional, qualified propositions: **X** is **F** is **a**;
- A conditional, unqualified proposition: If **X** is **F** then **Y** is **G**; and
- A conditional, qualified proposition: If **X** if **F** then **Y** is **G** is **a**.

where **X** and **Y** are fuzzy variables and **F** and **G** are fuzzy predicates or linguistic values (i.e., fuzzy sets on the domain of **X**) and **a** is a fuzzy probability value, such as *sort of true*, or *possible*. Here, a variable is defined as a fuzzy variable if its value is a fuzzy set on a universe of discourse. Any fuzzy set can be represented as a vector in which the value of each element gives the degree to which the corresponding member of the universe of discourse belongs to the fuzzy set that is the value of the fuzzy variable.

To support machine reasoning about fuzzy production rules of the above four types (i.e., *fuzzy reasoning*), three general inference principles have been proven to be particularly useful [14][16]. Although they are often used in sequence, a combination that involves an application of the particularization/conjunction principle followed by that of the projection principle is particularly effective. This combination is often referred to as *the compositional rule of inference* [13]. It includes as a special case a generalization of the *modus ponens*.

The main features of fuzzy reasoning, or approximate reasoning, which are relevant to the use and management of propositional vagueness, are the following:

- The truth-values are allowed to range over the fuzzy sets of T.
- The predicates used can be crisp or, more generally, fuzzy.
- The quantifiers used in fuzzy reasoning may be interpreted as fuzzy numbers which provide an imprecise characterization of the cardinality of one or more fuzzy or nonfuzzy sets.

As the values of a fuzzy variable used in a proposition are defined by fuzzy sets within universes of discourse, each fuzzy set serves to define the possibility that the variable could assume any particular value in that universe.

In fact the preceding paragraphs have implicitly outlined a new model for mechanized instructional planning, which is based on the concepts of fuzzy set and fuzzy reasoning. This model is intended to deal with the uncertainty/vagueness of the propositions in instructional strategic knowledge so that the fuzzy propositions or implications can be specified in terms of corresponding instructional operations and the inferences from them can be performed

To be more specific, a procedure for representing vague strategic knowledge and inferring specific decisions is given as follows:

1) Translate each proposition elicited from vague instructional strategic knowledge as one of the two types of fuzzy production rules (i.e., types (3) and (4) in Zadeh's theory).
2) Represent the predicates of the propositions as noncrisp denotations and the explicit and implicit modifiers as fuzzy or linguistic modifiers.
3) Specify the "universes of discourse" and the base values of fuzzy variables defined on them, such as the available instructional operations for certain instructional interventions and the important behavioral and cognitive variables involved in a student *model*. Define these fuzzy values in terms of fuzzy sets.
4) Perform fuzzy reasoning by applying a combination of

the fuzzy inference principles.

5) Set a threshold and select the base values from the fuzzy set outcomes whose grades of memberships are greater than it.

6) Activate corresponding instructional operations.

In what follows, we shall present the implementation of a prototype instructional planning component for ICAI systems, which employs the representation and reasoning technique mentioned above.

4. Implementing and Testing a Prototype ICAI Planning Component

The main purpose of implementing such a system is to demonstrate and test the feasibility of the proposed model for instructional planning. The prototype instructional strategies that were represented and used in this implementation are Collins and Stevens' inquiry teaching strategies. In [17], Collins and Stevens analyze the instructional dialogues used by some of the very best teachers they could find, and propose a theory of inquiry teaching based empirically on such teachers' work. In their theory, they have identified two major goals underlying most instructive dialogues; namely 1) teaching the use of a particular rule or theory in a given domain and 2) teaching how to develop a new rule or theory. Starting from these top- level goals, they further specify several sub-goals, which actually prescribe sub-optimal strategies for attaining the major goals.

In order to represent the strategic knowledge as revealed in Collins and Stevens' theory, we first reworded the key propositions in terms of condition-action fuzzy production rules. To a certain degree, those propositions stated in their theory specify the constraints of the inquiry teaching strategies to be applied. As a consequence, some constraint production rules were derived. Examples of the fuzzy production rules used in this implementation are given as follows:

Rule21 : If the student's knowledge about a particular rule or theory is *poor*, then the degree of teaching needed about that particular rule or theory is *high*;

Rule22 : If the student's skill at deriving a novel rule or theory is *poor*, then the degree of teaching needed about how to derive novel rule or theory is *sort of high*;

Rule23 : If the "instructor's" knowledge about the to-be-derived theory is *more or less poor*, then the degree of teaching about how to derive a rule or theory is *rather high*.

Rule24 : If the "instructor's" knowledge about the theory to-be-derived is *very good*, then the degree of teaching about the rule or theory is *high*.

In the implementation, the so-called "base values" anchor the various fuzzy variables in the fuzzy production system. Each fuzzy variable has a "universe of discourse", which gives the scale on which the underlying base values are defined. Since the value of a fuzzy variable is represented by a vector of eleven elements, each of the eleven dimensions will correspond to one base value - a point on a discretized psychological continuum. The real numbers which make up the value of the vector give the compatibility of each of the eleven base values on the scale appropriate to the particular context with the value of the fuzzy variable or constant. The elements of the "universe of discourse" used in this study are actually the indices of certain instructional tactics and/or operations, which are expected to be able to activate corresponding instructional sequences and/or operations as soon as they are selected. **Table 1** shows the base values for one of the variables used in the fuzzy production system.

Table 1
The base values for the fuzzy variable:
Degree of Teaching about How to Derive a Rule

a) Teach how to evaluate a rule or a theory:
   1) ask if a rule is correct or incorrect;
   2) ask if a value of dependent variable is correct;
   3) ask if factors are necessary or unnecessary;
   4) ask if factors are relevant or irrelevant;
   5) ask if the values of factors are correct;

b) Teach the form that a rule or a theory should take:
   6) give the structure of a rule or a theory;
   7) ask the student to construct different rules or theories of the idealized type;

c) Teach what questions to ask:
   8) ask the general question of what the value of dependent variable is;
   9) ask what necessary factors are;
   10) ask what relevant factors are; and
   11) ask what correct values of factors are.

The identification of these base values depended heavily on the analyses of the dialogue operations provided in the theory; and the assignments of grades for the fuzzy values such as *good* and *high* were based on Wenstop's definitions [18], but with some minor modifications. The definitions included in Wenstop's vocabulary have been proven to be a good approximation as far as the modelling of human language is concerned.

Having represented Collins and Stevens' inquiry teaching strategies as a set of fuzzy production rules and defined the "meaning" of each fuzzy value to be used, we developed an instructional planning component which would use such a representation. This planning component, along with its built-in reasoning and selection mechanisms, was then executed to generate sets of detailed instructional decisions.

During the first stage of execution, several interaction scenarios were created, in which the data would give the sim-

ulated inquiry teaching and learning situations. Accordingly the conditions in each scenario were summarized and grouped in terms of the base values of different fuzzy variables with their corresponding compatibility numbers. Thus each scenario was represented as a set of initial values of certain fuzzy variables, which were fuzzy values.

As in the second stage of the planning process, fuzzy reasoning was performed based on each set of the initial data. In this stage, some fuzzy production rules might be fired and the consequences of this would provide suggestions for the instructional interventions in fuzzy or linguistic terms. For example, one rule might suggest that in certain circumstances, *the degree of teaching how to derive a rule or theory* should be *sort of high*.

The control mechanism for the planning component controls, selects, and evaluates rules to be manipulated during the process of approximate reasoning. The outcomes from the reasoning module are fuzzy sets. Therefore by weighing all possible operations and selecting some of them, the planning component would output specific instructional decisions and thus activate corresponding operations.

In order to test the usefulness of the proposed representation and reasoning technique, the results of 100 decision rules inferred from the execution were evaluated with respect to those stated by the human instructors. As a result of the comparison of large quantities of data obtained by feeding in the test case situations from [17]and seeing if the prescribed output corresponded to the choices actually made by the original model teachers, it was concluded that the decisions generated by the planning component were accurate enough to model the rules used by human experts for ICAI inquiry- teaching purposes.

5. Conclusions

In this paper, we have described a new approach to planning instruction in ICAI systems, using only minimum vague instructional strategic knowledge. This approach incorporates fuzzy production rule representation with approximate reasoning to infer and select specific instructional decisions.

References

[1] Carbonell, J.R. (1971a). AI and CAI: An artificial intelligence approach to computer-aided instruction, *IEEE Transactions on Man-Machine Systems MMS-11(4)*.

[2] Hartley, J.R., and Sleeman, D.H.(1969). Towards more intelligent teaching systems. *International Journal of Man Machine Studies, Vol. 1*. pp.177-188.

[3] Carbonell, J.R. (1970b). Mixed-initiative man-computer instructional ialogues. *BBN Rep. No. 1971*, Bolt Beranek and Newman, Inc., Cambridge, Mass.

[4] Stevens, A.L., and Collins, A. (1977). The goal structure of a socratic tutor. *BBN Report No. 3518*, Bolt Beranck and Newman, Inc., Cambridge, Mass.

[5] Park, O-K, Perez, R.S., and Seidel, R.J. (1987). Intelligent CAI: Oldwine in new bottles, or a new vintage? in Greg P. Kearsley (Ed.), *Artificial Intelligence and Instruction: Applications and Methods*. Addison-Wesley, Mass.

[6] Bruner, J.S. (1966). Theorems for a theory of instruction, in J. Bruner (Ed.), *Learning About Learning: A Conference Report*, U.S. Government Printing Office, Washington, D.C..

[7] Goldstein, I. (1979). The genetic epistemology of rule systems. *International Journal of Man-Machine Studies, Vol. 11*.

[8] Self,J.(1985). A perspective on intelligent computer-assisted learning. *Journal of Computer Assisted Learning, Vol. 1(3)*, PP.149-158.

[9] Collins, A., and Stevens, A.L. (1982). Diagnosing student's misconceptions in causal models, in D. Sleeman, and J.S. Brown (Eds.), *Intelligent Tutoring Systems*. Academic Press, Cambridge, Mass.

[10] Brown, J.S., and Burton, R.R. (1979). An investigation of computer coaching for informal learning activities, in *International Journal of Man-Machine Studies , Vol. 11*.

[11] Hartel, G.D., Walberg, H.J., and Weinstein, T. (1983). Psychological models of educational performance: A theoretical synthesis of constructs. *Review of Educational Research, Vol. 53*.

[12] Zadeh, L.A. (1965). Fuzzy sets. *International Journal of Control, Vol. 8*.

[13] Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics Vol. SMC-3, No.1*.

[14] Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning. *Information Science , Vol. 8*

[15] Zadeh, L.A. (1981). PRUF - A meaning representation language for natural language in E.H. Mamdani, and B.R. Gaines (Eds.), *Fuzzy Reasoning and its Applications*. Academic Press, London

[16] Zadeh, L.A. (1985). Fuzzy reasoning, in M.M. Gupta, et al. (Eds.), *Approximate Reasoning in Expert Systems*, North-Holland, Amsterdam.

[17] Collins, A., and Stevens, A.L. (1982). Goals and strategies of inquiry teachers, in R. Glaser (Ed.), *Advances in instructional Psychology, Vol. 2*, Lawrence Erlbaum Associates, Hillsdale, N.J..

[18] Wenstop, F. (1976). Deductive verbal models of organizations. *International Journal of Man-Machine Studies, Vol. 8*.

[19] Woolf, B., Murray, T., Suthers, D., and Schultz, K. (1988). Knowledge Primitives for Tutoring Systems, in *Proceedings of the 1988 International Conference on Intelligent Tutoring Systems, Montreal*.

COMPUTER SUPPORTED PEER EDUCATION

J. Swallow
Dept. of Education
Concordia University
Montreal, PQ

A study was conducted to research and develop an environment in which students acquire the strategies of educators. Peer education strategies are employed by students to draw out knowledge from their peers. This knowledge is then used to make connections between ideas and to diagnose misconceptions. The computer provides support for peer education in four ways. The first is by cueing peer educators to look for statements made by their peers which could be profitably pursued. The second is by providing peer educators with questions designed to analyze these statements. The third is by providing peer educators with a list of possible responses to their peers' statements. The fourth is by cueing peer educators to evaluate the success of the strategies employed. The study involved students in grade 6. The strategy learning demonstrated by computer supported peer educators was compared with strategies employed by unsupported peer helpers. The results indicate that computer supported peer educators were more likely to ask their peers to engage in activities which required reflective thought. Requests were also more likely to be open ended. In addition, computer suported peer educators were less likely to lecture their peers. This resulted in more active learning on the part of the peers of computer supported peer educators. It also resulted in extended discussions of ideas generated by the students themselves.

ENSEIGNEMENT MUTUEL ASSISTÉ PAR ORDINATEUR

On a mené une étude pour rechercher et concevoir un environnement dans lequel les étudiants acquerraient les stratégies d'enseignants. Les étudiants se servent des stratégies de l'enseignement mutuel pour tirer des connaissances de leurs pairs. Ces connaissances sont ensuite utilisées pour établir des liens entre les idées et déceler les conceptions erronées. L'ordinateur apporte son soutien de quatre façons à l'enseignement mutuel. D'abord, il incite les étudiants à rechercher les énoncés de leurs pairs auxquels il pourrait être utile de donner suite. Deuxièmement, il fournit aux étudiants des questions qui ont pour objet d'analyser leurs énoncés. Troisièmement, l'ordinateur fournit aux participants de l'enseignement mutuel une liste des réponses qu'ils peuvent donner aux énoncés de leurs pairs. Enfin, il incite les participants à évaluer l'efficacité des stratégies utilisées. L'étude a porté sur des élèves de 6^{e} année. On a comparé une stratégie employée par les participants à l'enseignement mutuel assisté par ordinateur à celle utilisée par des étudiants qui ne se sont pas servis de l'ordinateur. Les résultats ont démontré que les participants à l'enseignement mutuel assisté par ordinateur étaient plus enclins à demander à leurs pairs d'entreprendre des activités exigeant une reformulation. Les demandes étaient également plus susceptibles d'être évolutives. En outre, les participants à l'enseignement mutuel assisté par ordinateur avaient moins tendance à faire des exposés didactiques à leurs pairs. Par conséquent, les pairs des participants à l'enseignement mutuel assisté par ordinateur avaient une formation plus active et des discussions plus poussées sur des idées qu'ils avaient eux-mêmes exprimées.

COMPUTER SUPPORTED PEER EDUCATION

J. Swallow
Dept. of Education
Concordia University
Montreal, PQ

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

RÉÉDUC: UN PROGRAMME DE RÉÉDUCATION COGNITIVE SUR MICRO-ORDINATEUR

M. Pépin, S. Larose
École de psychologie
Université Laval
Québec, PQ

Les programmes d'intervention en rééducation cognitive sont principalement basés sur la répétition d'exercices visant à stimuler des habiletés cognitives de base. Dans ce contexte, le micro-ordinateur s'avère un outil utile qui peut à la fois permettre la présentation d'exercices et la gestion du cheminement d'un sujet à travers les exercices disponibles. RÉÉDUC est un logiciel offrant 8 types d'exercices visant les habiletés-cibles suivantes: attention, mémoire, coordination main-oeil, visualisation spatiale, orientation spatiale, poursuite visuelle, vitesse perceptuelle. Ce logiciel permet un cheminement individualisé en fonction des difficultés des sujets. Le logiciel s'adresse particulièrement aux jeunes montrant des symptômes associés à la dysfonction cérébrale mineure, de même qu'aux traumatisés crâniens. Nous présenterons diverses études visant à démontrer l'efficacité de RÉÉDUC. Notre communication mettra également en évidence les composantes du programme et son fonctionnement selon un mode "adaptatif".

REEDUC: A COGNITIVE REEDUCATION PROGRAM ON A MICROCOMPUTER

Programs used as cognitive reeducation measures are primarily based on the repetition of exercises intended to stimulate basic cognitive abilities. In this context, the microcomputer is proving a useful tool that allows both the presentation of exercises and control of the guidance of subjects through the available exercises. REEDUC is a software package offering eight types of exercise aimed at the following target abilities: attention, memory, hand-eye co-ordination, spatial visualization, spatial orientation, visual tracking and speed of perception. With this software, a personalized path is possible as a function of the difficulties of subjects. The software is intended particularly for young people showing symptoms associated with minor cerebral dysfunction, as well as victims of cranial trauma. I will describe various studies designed to demonstrate the effectiveness of REEDUC. My paper will also discuss the components of the program and its operation in an "adaptive" mode.

REEDUC: Un programme de rééducation cognitive sur micro-ordinateur

Pépin Michel et Larose Simon

Université Laval

PROBLEMATIQUE

C'est tout récemment que la psychologie a commencé a s'intéresser aux nombreuses possibilités qu'offre le micro-ordinateur en rééducation cognitive.

L'avènement du micro-ordinateur a d'abord suscité de nombreuses recherches se rapportant à ses utilisations purement récréatives. Ces études traitent des propriétés motivationnelles de ces jeux informatisés (Malone, 1980) et des habiletés cognitives sollicitées par la pratique de ceux-ci (Chaffin, Maxwell & Thompson, 1982; Dorval et Pépin, 1986; Driskell & Dwyer, 1984; Gagnon, 1985; Greenfield, 1983; Griffith, Voloschin, Gibb & Bailey, 1983; Jones, 1984). Sur ce dernier thème plusieurs chercheurs s'entendent à dire que l'on retrouve à l'intérieur de plusieurs jeux informatisés des composantes spatiales, attentionnelles et psychomotrices (Lowery & Knirk, 1982; Loftus & Loftus, 1983; Greenfield, 1984; Jones, 1984). La relation entre les performances à des tests d'habileté spatiale et celles à des jeux et exercices informatisés a été démontrée pour les variables de poursuite visuelle (Jones, Kennedy et Bittner, 1981) de balayage visuel et de coordination motrice (Griffith et al., 1983), de visualisation spatiale (Gagnon, 1985) et de relations spatiales (Larose, Gagnon, Ferland et Pépin, 1989).

Considérant leur fort contenu spatial, plusieurs études ont vérifié l'effet d'une utilisation appropriée des jeux et exercices informatisés sur le développement de certaines habiletés cognitives de sujets normaux. Les résultats appuient fermement l'hypothèse que ce médium permet de stimuler la visualisation spatiale (Dorval et Pépin, 1986; Gagnon, 1985; McClurg et Chaillé, 1987; Barsam et Simutis, 1984), l'orientation spatiale (Forsyth, 1986) ainsi que l'exploration et la poursuite visuelle (Larose et al., 1989). Le micro-ordinateur, par ses capacités visuelles (ex.: mouvement bidimensionnel et tridimensionnel), ses possibilités rétroactives et ses facilités techniques (ex.: enregistrement de la performance, utilisation individualisée) devient un outil avantageux pour des fins de rééducation. Quelques études menées auprès d'individus présentant des troubles d'apprentissage et des dommages cérébraux d'origine traumatique ou lésionnelle ont suggéré l'efficacité du micro-ordinateur comme complément à l'intervenion thérapeutique (Ferland, 1988; Lynch, 1981, 1983a, 1983b; Sivak, Hill & Olson, 1984).

Lynch (1983), directeur du Brain Injury Rehabilitation Unit du Veterans Administration Medical Center de Palo Alto, fut l'un des premier à faire appel aux jeux informatisés pour pallier à certains déficits moteurs et spatiaux de ses patients. A partir d'une pratique régulière du jeu informatisé Super Breakout chez douze sujets présentant une lésion cérébrale, il note une amélioration de la performance maintenue jusqu'à six semaines après l'arrêt de l'entraînement.

Bracy (1983) a vérifié les qualités thérapeutiques d'un programme d'entraînement informatisé auprès d'une clientèle d'accidentés cérébraux. La tâche contenait des exercices visuo-spatiaux, des exercices de coordination motrice et des épreuves d'orientation. L'évaluation des sujets montre une amélioration importante des habiletés en cause. De leur côté, Beattie et Owen (1985) obtiennent des résultats quelque peu différents. Sur trois patients accidentés cérébraux, un seul montre une amélioration dans une épreuve visuo-spatiale alors que tous montrent une amélioration importante dans une épreuve de mémoire.

Larose et al. (1989) ont vérifié l'effet de douze heures d'entraînement du jeu vidéo Super Breakout auprès de soixante enfants en difficultés d'apprentissage. La majorité d'entre eux présentent une dysfonction cérébrale mineure (DCM). Cette population fut retenue par les auteurs parce qu'elle montre de faibles habiletés spatiales. Une analyse multivariée pratiquée sur les gains de poursuite visuelle et de balayage visuel montre que l'amélioration est plus considérable dans le groupe expérimental que dans le groupe témoin. Les auteurs suggèrent que la spécificité de la tâche est un facteur important dans l'efficacité de l'entraînement. Le Super Breakout est un jeu simple et il est facile d'y identifier les habiletés en jeu. Selon les auteurs, l'élaboration d'exercices simplifiés devient avantageuse car ceux-ci présentent une validité plus grande avec les habiletés entraînées.

REEDUC: OBJECTIFS ET CONTENU GENERAL

C'est avec l'objectif de profiter des avantages de la micro-informatique et d'améliorer l'intervention thérapeutique en rééducation cognitive que nous avons développé et évalué une batterie d'exercices informatisés. REEDUC est destinée aux personnes présentant des déficits fonctionnels au niveau des habiletés cognitives impliquées dans tout apprentissage. Elle propose huit exercices différents permettant de stimuler les habiletés cognitives suivantes: attention, mémoire visuelle, coordination main-oeil, poursuite visuelle, vitesse perceptuelle, visualisation et orientation spatiale, discrimination figure/fond et reconnaissance de formes. Ces habiletés de base sont souvent impliquées dans plusieurs apprentissages. La lecture et l'écriture font appel, entre autres, à l'exploration visuelle, la poursuite visuelle et la discrimination figure/fond. Les notions de géométrie sont directement reliées aux habiletés d'orientation et de visualisation spatiale. Ces habiletés sont souvent déficitaires chez les enfants ayant des difficultés d'apprentissage.

Ce logiciel hautement interactif fonctionne selon un mode "adaptatif" qui permet à chaque utilisateur de progresser à son propre rythme. Chaque exercice est simple et vise le développement d'une ou de quelques habiletés. L'apprentissage se fait par niveaux de difficulté. Toutes les tâches débutent avec des item relativement faciles. Les item deviennent plus difficiles selon l'apprentissage de l'utilisateur. Des structures de renforcement permettent de faciliter l'apprentissage de l'utilisateur. La gestion de la batterie se fait par l'intermédiaire de menus simples.

REEDUC est un outil qui permet aux pédagogues et autres professionnels des milieux éducatifs d'intervenir plus efficacement et de stimuler l'apprentissage dans un contexte intéressant.

ETUDE PRELIMINAIRE

Récemment, nous avons mené une étude pour vérifier l'hypothèse qu'une pratique continue et régulière des exercices informatisés de la batterie REEDUC peut développer l'habileté spatiale telle que mesurée par une série d'épreuves psychométriques. Soixante-cinq enfants de quatrième et cinquième année, tous présentant des troubles variés d'apprentissage, ont participé à l'étude. Parmi ces enfants, certains affichent des symptômes d'hyperactivité, d'autres sont identifiés clairement comme ayant une dysfonction cérébrale mineure. Une troisième partie de l'échantillon montre des troubles attentionnels sans hyperactivité. Tous ont des troubles d'apprentissage importants qui affectent leur acuité spatiale.

Variables dépendantes

Cinq instruments ont permis de mesurer les dimensions généralement rapportées comme formant le construit spatial.

- Minnesota Paper Form Board (Likert & Quasha, 1970)
- Le sous-test de Reconnaissance de formes du BGTA (Batterie générale de tests d'aptitudes)
- Le test d'Aptitudes Mentales Primaires (Thurstone & Thurstone, 1963)
- Le test de Barrage A-C (Diller, 1974)
- Le Symbol Digit Modalities Test (SDMT) (Smith, 1973)

Ces tests mesurent dans l'ordre, la visualisation spatiale, la reconnaissance de formes, les relations spatiales, l'exploration visuelle et la mémoire visuelle.

Procédure

L'étude fut réalisée à partir d'un devis quasi-expérimental avec groupe témoin. Tous les sujets ont été rencontrés une première fois pour compléter les cinq épreuves psychométriques. Quarante-sept de ces enfants se sont entraînés pendant vingt heures avec les exercices de la batterie REEDUC. L'entraînement fut intégré au programme scolaire des enfants. Les jeunes des groupes expérimental et contrôle ont été rencontrés une semaine après la fin de l'entraînement pour l'évaluation du post-test. Le déroulement fut identique à celui du prétest.

Résultats

Une analyse multivariée, effectuée sur les performances aux tests d'habileté spatiale, a montré que le groupe ayant participé au programme REEDUC obtient des gains légèrement supérieurs au groupe témoin et ce, à quatre des cinq épreuves psychométriques (T de Hotellings = .27, $F(5,43)=2.32$, $p = .059$). Comme l'indique la fonction discriminante présentée au tableau I, cette amélioration est surtout due aux variables d'exploration visuelle et de reconnaissance de formes qui présentent les poids les plus importants.

TABLEAU I

FONCTION DISCRIMINANTE QUI EXPLIQUE L'ECART ENTRE LE GROUPE EXPERIMENTAL ET LE GROUPE CONTROLE

.14visualisation + .21reconnaissance de formes - .10orientation + .84exploration visuelle + .21mémoire visuelle

Discussion

Ces résultats montrent qu'un entraînement approprié à l'aide d'exercices informatisés peut stimuler le développement de certaines fonctions cognitives de clientèles en difficultés d'apprentissage. Une analyse plus nuancée des résultats de cette étude préliminaire nous a permis de tirer deux considérations importantes:

1) Ce type de programme permet davantage le développement d'habiletés visuo-spatiales de base que l'on retrouve souvent dans des contenus académiques particuliers. C'est le cas de l'exploration visuelle, la reconnaissance de formes géométriques et la mémoire visuelle. L'effet sur les habiletés d'orientation et de visualisation spatiale est moins important possiblement à cause de la complexité des processus requis pour la résolution de ce type d'habiletés spatiales. Pour favoriser le développement d'habiletés plus complexes, il serait pertinent de planifier un entraînement séquentiel. Les enfants devraient alors réussir parfaitement les exercices de reconnaissance de formes, d'exploration visuelle et de mémoire visuelle avant de s'attaquer aux exercices

de visualisation et d'orientation spatiale.

2) le programme est surtout efficace pour les enfants qui présentent des troubles attentionnels aigus. Les enfants hyperactifs ont été ceux qui ont le moins bénéficié de l'entraînement. Sachant que cette clientèle présente un grand besoin de changement et d'activation environnementale, il devient plus difficile de prolonger la pratique de certaines tâches. Ce fut le cas des épreuves de visualisation et d'orientation spatiale. Sans doute qu'un suivi individuel serait beaucoup plus bénéfique pour atteindre l'objectif de développer les fonctions spatiales chez les enfants hyperactifs. Leur attention étant beaucoup plus affectée, il se peut que ces enfants aient besoin de plus de temps pour bien intégrer les habiletés.

CONCLUSION

Une application individualisée de la batterie visuo-spatiale pourrait faciliter le travail des aides pédagogiques et des psychologues oeuvrant dans les milieux scolaires. Bien que la vocation de REEDUC est de développer les aptitudes cognitives, son utilisation comme mesure psychométrique peut être envisagée. Ainsi, cette initiative permettrait de munir les éducateurs, les enseignants et les professionnels scolaires d'un outil de dépistage avantageux qui vise des cibles d'intervention spécifiques.

REFERENCES

Berman, H. F. & Simutis, Z. M. (1984). Computer-based graphics for terrain visualization training. Human Factors, 26(6), 659-665.

Beattie, R. L. & Owen, N. (1985). Preliminary study of cognitive retraining via computer-based activities. Perceptual and Motor Skills, 61, 1130.

Bracy, O. L. (1984). Using computers in neuropsychology. In M. D. Schwartz (Ed.), Using computers in clinical practice (pp. 257-264). New-York: Hayworth Press.

Bracy, O. L. (1983). Computer based cognitive rehabilitation. Cognitive rehabilitation, 1, 7-8 et 18.

Chaffin, J. D., Maxwell, B., & Thompson, B. (1982). ARC-ED curriculum: The application of video game formats to educational software. Exceptional Children, 49, 173-178.

Dorval, M. et Pépin, M. (1986). Effect of playing a video game on a measure of spatial visualization. Perceptual and Motor Skills, 62, 159-162.

Driskell, J. E. & Dwyer, D. J. (1984). Microcomputer video game based training. Educational Technology, February, 11-15.

Ferland, C. (1988). Effet de la pratique d'un jeu informatisé sur le développement des habiletés visuo-spatiales chez des enfants présentant une dysfonction cérébrale mineure. Thèse de doctorat inédite. Université Laval, Québec.

Forsyth, A. S. (1986). Applications of microcomputer search-type adventure games in research on spatial orientation, place location, and memory for places. Paper presented at the annual meeting of the american educational research association, San Francisco. (Document ERIC) reproduction #ED 273 264, 12 pp.)

Gagnon, D. (1985) Videogames and spatial skills: an exploratory study. Educational communication and technology: Journal of theory, research and development, 4, 263-275.

Griffith, J. L., Voloschin, P., Gibb, G. D. & Bailey, J. R. (1983). Differences in eye-hand motor coordination of video-game users and non-users. Perceptual and Motors Skills, 57, 155-158.

Jones, M. B., (1984). Video games and psychological tests. Simulation and games, 15, 131-157.

Jones, M. B., Dunlap, W. P. & Bilodeau, I. McD (1986). Comparison of video game and convention test performance. Simulation and Games, 17, 435-446.

Klein, E. L. (1985). Computer delivery of memory retraining with head injured patients. Dissertation Abstracts International, 46, 305-B.

Larose, S., Gagnon, S. Ferland, C. & Pépin, M. (1989). Spatial training through computer games. Perceptual and Motor Skills (sous presse).

Loftus, G. R. & Loftus, E. F. (1983). Mind and Play. New-York: Basic books inc.

Lowery, B. R. and Knirk, F. G. (1982). Micro-computer video games and spatial visualization acquisition. Journal of education technology systems, 11 155-166.

Lynch, W. J. (1983). Video games in remediation. Proceedings of a Symposium on Video Games and Human Development (pp. 25-28). Cambridge, MA.

McClurg, P. A. & Chaillé, C. (1987). Computer games: environments for developing spatial cognition? Journal Educational Computing Research, 3, 95-111.

Malone, T. W. (1981). What makes computer games fun? Byte, 6, 258-277.

Pépin, M. (1988). REEDUC (batterie d'habiletés spatiales). Québec: Université Laval, Ecole de Psychologie.

Sivak, M., Hill, C. S., Olson, P. L. (1984). Computerized video tasks as training techniques for driving-related perceptual deficits of persons with brain damage: a pilot evaluation. International Journal of Rehabilitation and Research, 7, 389-398.

Lynch, W. J. (1983b). Cognitive retraining using microcomputer games and commercially available software. Cognitive Rehabilitation, 1, 19-22.

Lynch, W. J. (1981). TV games as therapeutic interventions. In W. J. Lynch, Rehabilitation of post-traumatic brain-damaged patients. Symposium of American Psychological

Association, Los Angeles, CA.

Jones, M. B., Kennedy, R. S. & Bittner, A. C. (1981). A video game for performance testing. American Journal of Psychology, 94, 143-152.

DEVELOPMENT AND STUDENT USE OF A MICROCOMPUTER-BASED "SMART BOOK" CONCERNING SEXUALLY TRANSMITTED DISEASES

M.J. Brody, D. Kopec
Science Education
University of Maine
Orono, ME, USA

This paper describes the development of a microcomputer-based instructional system dealing with sexually transmitted diseases (STDs) in college age populations. The concept of a "smart book" has been employed and incorporates three components: (1) a graphic representation of expert knowledge in a semantic network, (2) a hypertext information module and (3) an expert system. The instructional technology has been deployed on a Macintosh local area network in the student union at the University of Maine and is presently designed to provide information to undergraduates on six STDs including AIDS. Information concerning STDs was first elicited from medical doctors and health educators specializing in STDs through the use of group interviewing and concept mapping techniques. The results were summarized by the researchers into semantic networks using the Apple HyperCard application. For each node in the network corresponding textual information concerning individual STD concepts was collected and linked together using superordinate concepts which relate different areas of the network. Finally an expert system was designed to give students advice about where in the network information relevant to their needs was located. The combination of these three elements provides us with the paradigm of a "smart book" which helps student locate information in a more efficient and meaningful way.

DÉVELOPPEMENT ET USILISATION PAR LES ÉTUDIANTS D'UN GUIDE INTELLIGENT PILOTÉ MICRO-ORDINATEUR SUR LES MALADIES TRANSMISES SEXUELLEMENT

Le présent article décrit le développement d'un système didactique informatisé traitant des maladies transmises sexuellement (MTS) chez les étudiants de niveau collégial. Le concept de "guide intelligent" utilisé comporte trois éléments : (1) la représentation graphique de connaissances expertes dans un réseau sémantique, (2) un module d'information hypertexte et (3) un système expert. La technologie d'enseignement a été déployée dans un réseau local Macintosh mis en place au local de l'association des étudiants de l'Université du Maine, et est actuellement conçue pour informer les étudiants de premier cycle sur six MTS, y compris le SIDA. Les renseignements concernant les MTS ont d'abord été tirés de médecins et d'éducateurs en hygiène spécialistes dans ce domaine au cours d'échanges de groupes et à l'aide de techniques de rapprochement de concepts. Les chercheurs professionnels ont présenté les résultats sous forme de réseaux sémantiques utilisant l'application de HyperCard Apple. Pour chacun des noeuds du réseau, on a recueilli et groupé des textes correspondants portant sur des notions MTS individuelles, en utilisant des concepts pères qui relient différentes zones du réseau. Enfin, un système expert a été créé pour aider les étudiants à repérer les renseignements dont ils ont besoin dans le réseau. La fusion de ces trois éléments nous donne le paradigme d'un "guide intelligent" dont l'objet est d'aider les étudiants à trouver les renseignements de façon plus efficace et plus constructive.

Development and Student Use of a Microcomputer-based "SMART BOOK" Concerning Sexually Transmitted Diseases.

Michael Brody, Danny Kopec

University of Maine
Orono, Maine 04469

ABSTRACT

This paper describes the development of a microcomputer-based instructional system dealing with sexually transmitted diseases (STDs) in college age populations. Information concerning STDs was first elicited from medical doctors and health educators specializing in STDs through the use of group interviewing and concept mapping techniques. The results were summarized by the researchers into semantic networks using Apple's HyperCard application. For each node in the network corresponding textual information concerning individual STD concepts was collected and linked together using superordinate concepts which relate different areas of the network. The combination of these elements provides us with the paradigm of a "smart book" which helps student locate information in a more efficient and meaningful way.

The concept of a "smart book" incorporates two components; (1) a graphic representation of expert knowledge in a semantic network and (2) a hypertext information module. Students in the University of Maine School of Nursing have been interacting with the STD "smart book" in order to study concepts and help develop clinical decision making skills. Student use has been monitored through individual interviews during interaction with the program. The analysis focuses on how students progressed through the program and why they would make specific choices in the network. Preliminary results indicate that the program is effective in providing relevant information on STDs and will help guide the design of an intelligent tutoring system.

Introduction

Sexually Transmitted Disease (STD) and especially the threat of AIDS is widely recognized as an important educational problem, particularly in the college age population. To date, intelligent software has not been developed and employed to effectively disseminate the critical knowledge relevant to this epidemic. Since STDs as a group are a personal problem which few people feel comfortable discussing, we believe the computer can provide an opportunity for learning about these diseases while insuring anonymity. This paper describes the preliminary steps we have taken to begin the development of an STD intelligent tutoring system (STD/ITS) for college age students.

The principle focus of our research is in the areas of knowledge acquisition, knowledge representation, learning, and problem solving with respect to the identification, control, and treatment of STDs. Fundamental issues which we address are (1) the effective extraction, representation, and modification of expert/novice medical knowledge,(2) the design of an effective computer based information system for STDs and (3) the acquisition of problem solving techniques used by novice learners to apply this knowledge base within the context of real world needs.

This paper describes our work with domain experts using concept maps and group interviews to construct the STD knowledge base, the development within HyperCard of an STD "smart book", undergraduate nursing students use of the STD "smart book" and finally implications for the construction of an STD intelligent tutoring system.

Prior Work

Considerable effort has been expended towards study in the area of cognitive skills, their acquisition, and their modelling with such systems as GPS (Newell & Simon, 1972), HAM (Anderson & Bower, 1973), and ACT (Anderson, 1976), but the principle problems which still remain are (a) what is knowledge (advice), (b) how should it be represented, (c) how can it be effectively be extracted and represented for novice learners?

It has been determined that there are essentially three areas of skill/knowledge acquisition: encoding, proceduralization, and composition (Anderson, 1982). Furthermore, the knowledge associated with such skill acquisition tends not to be constructed in a formal top-down manner. Experts seem to be able to perform cognitive leaps based upon acquired know-how or intuition. However, it is accepted that experts tend to be poor in articulating precisely just how they are able to solve difficult problems within their domains of expertise.

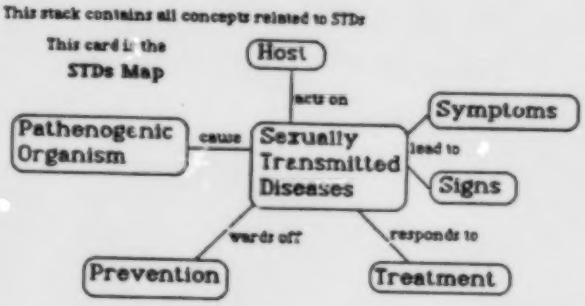
Concept Maps as Knowledge Representation

Research in knowledge representation has come from three distinct fields: AI, linguistics, and cognitive psychology. Work in linguistics provides us with the basic premise that knowledge can be reduced to a relatively small set of relations between concepts. Work in AI, based on a cognitive psychological perspective, provides us with techniques for knowledge representation. One such approach being to combine small subsets of knowledge into larger networks connected by relations, i.e., semantic networks. Quite a few researchers in cognitive science have adopted these methods to help illustrate thinking and learning.

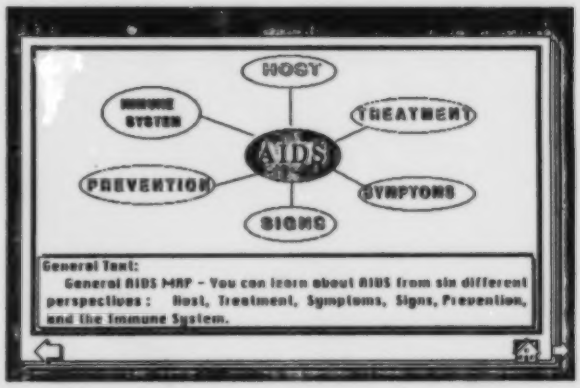
In this study we utilize a graphical knowledge representation methodology called concept mapping which has been successfully applied to a variety of formal classroom learning situations (Novak and Gowin, 1984) and informal adult education (Brody, 1985). Concept Mapping has previously been somewhat independent of the AI domain intended here for its application. However, its roots in cognitive psychology and learning suggest its suitability for our investigations. In addition, we know of at least one other ongoing research effort which is based on this form of knowledge representation (Feifer, Dyer, and Baker, 1988).

Concept mapping provides a framework for representing subject/expert/novice knowledge. It is based on the assumption that we think with concepts and that these ideas are liked together in meaningful idiosyncratic relationships by the learner. Large numbers of concepts and their relations are joined together into a major conceptual framework or schema. Concept maps of small numbers of concepts can then be joined together forming complex semantic networks representing schema (Collins and Loftus,1975).

In this study concept maps were used to help extract expert and novice knowledge, identify key concepts involved in STDs, and represent them in easily understandable formats for both future teaching and learning in a computer-based intelligent tutoring system. University of Maine Health Center medical doctors and health educators collaborated on the construction of concept maps for six STDs during group interviews. The resulting maps help make clear the relatively small number of concepts and relations that need to be focused upon in learning about STDs. The figure below is a concept map of the major subsuming concepts in an extensive network of maps related to STDs.



Concept maps were constructed for six STDs by the campus health care experts and collectively represent their understanding of the infectious diseases. A number of critical concepts; signs symptoms, prevention, treatment, host, and transmission provide a pervading skeletal structure for each disease. This forms the basis of a semantic network knowledge base from which we plan to construct an intelligent tutoring system's expert and student model components.



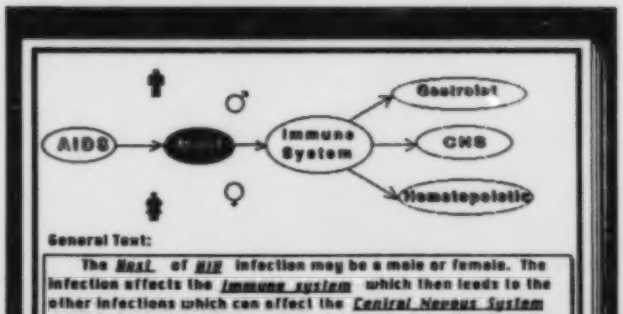
The "Smart Book"

The issues of knowledge representation and acquisition are fundamental to the study of artificial intelligence and intelligent tutoring. Most people would probably agree that much formal learning relies on books consisting of words which comprise textual information or photos which comprise graphical information. A well-written and well-structured book can be considered an excellent source of information on a subject and integral for a student's learning. Books tend to be complete and sequential in their presentation of material. There is also an implicit hierarchical structure for the knowledge in

most books which attempt to present learning material. This structure entails the presentation of material in a linear and top-down form. That is, the general overview of a subject is presented first and then details on specific relevant topics or methods will follow. Each topic is presented in an order prescribed by the author. A good text will have all the important subject information as well as a table of contents and index detailed enough to direct the reader to information on a topic of interest covered in the book. Key words may also be highlighted in some way.

However, lacking in traditional books is a flexibility in the order of presentation of information and an omission of important relationships between concepts presented in various parts of the book. This is due to the very static nature of this form of knowledge representation, the style and content being rigid and unchangeable once a book is published.

The primary advantage of our "smart book" is flexibility and the opportunity for the learner to explore relationships between concepts. That is, it can be used in many ways and important links between concepts are included in the presentation. The order in which material to be learned is presented is the choice of the user. In essence, the "smart book", as coined by Feigenbaum (1988), represents a road map through the knowledge base of STD's.



Information in the "smart book" is represented in two forms: graphically and textually. Graphical information has been derived from a form of knowledge representation called *concept maps*. The structure of these maps can embellish the knowledge of experts in a domain. Typically any node on a screen can be "clicked" to proceed to the next screen with a new map segment and more information. The key to a "smart book's" flexibility is that one can move in many directions via the nodes and arcs in a graph. Concepts in nodes are connected by arcs. Importantly, at all times the user can quickly determine how the current node was reached and what are the possibilities for proceeding from the current node. Textual information is always presented in a brief, compact and clear form.

The system has been developed in Hypercard, the Macintosh hypermedia system, which enables the user interface with many forms of information. Transparency in form and function is fundamental to the "smart book". In addition to existing pop up windows, there is the potential for linking to a glossary of terms, synonyms for key words, a retrace facility, expert advice, and video-disk presentation of graphical information. As any good knowledge base, it is easy to modify, expand, and refine. The "smart book" is viewed as an important stepping stone to ultimate goal of building the STD/ITS.

Use of the STD "Smart Book" with Learners

Presently we are using the STD "smart book" (focusing on Aids) with undergraduate nursing students. We are interested in how they access the information and how they interact with the system.

A general introduction to the Macintosh computer emphasizing icons, pull down menus and point and click techniques was presented to an entire class. The students will be asked to spend at least two sessions working on the STD "smart book" and approximately 12 students will be interviewed as they are using the system.

The paths which the students take through the "smart book" will be recorded. Interviewers will ask student users specific questions about their understanding of the material presented on screen as well as their feelings about their interactions with the system.

It is our expectation that the flexibility of the "smart book" approach will enable the students to learn in idiosyncratic ways and thus enhance meaningful

learning about STDs.. We also believe that because of the rapid access to interconnections between various concepts that the students will appreciate the systems ability to help them process information quickly and help facilitate learning clinical decision making processes.

Towards an Intelligent Tutoring System

Recent advances in learning have taken a particularly cognitive perspective (Carey, 1986) and many of the major findings have implications for education in general and the development of intelligent tutoring systems in particular. This perspective begins with the principle that what learners already know about new information will influence their subsequent learning of that material (Champagne and Klopfer, 1984). From this point of view the ways in which undergraduate nursing students access the STD "smart book" will be compared with how the medical experts constructed the original STD semantic networks. This information in integral to the construction of a student modeler in the STD/ITS since student models must be compared to expert understanding in order for the tutor to make suggestion for subsequent learning events.

In addition to refining the concept of the STD "smart book",we are working on the implementation of an expert system based on the medical information embedded in the STD semantic network. It is our expectation that the expert system will be used in the STD/ITS in order to help the learners understand the relationship of general STD concepts to specific diseases. For example, after a learner has traversed the "smart book" and assimilated some knowledge she will be able to ask the expert what types of devices can be used to prevent infection. The expert will then lead the student through an analysis of types of diseases, sign, symptoms and subsequently prevention techniques.

The future construction of an intelligent tutoring system to represent knowledge bases of both experts and novices will allow the researchers to assess learners' knowledge and how that knowledge changes over time. Further studies will entail human subject protocol analysis of interactions with the STD "smart book" and a comparison of the difference in levels of understanding following lecture, text and "smart book" instruction.

Our prototype system will serve as a testbed for investigation into a number of well known problem solving paradigms and in particular the acquisition of clinical decision making skills. Its ability to transmit information about STD detection, control and abetment will provide insight into future techniques for incorporating both expert and novice knowledge into existing knowledge bases.

References

Anderson, J. R., and Bower, G. H. (1973) Human Associative Memory. Washington, DC: Winston.

Anderson, J. R. (1976) Language, Memory, and Thought. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J.R. (1982) Cognitive Skills and Their Acquisition. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J. R. (1982) Acquisition of Cognitive Skill. Psychological Review, vol. 8, no. 4,pp. 369-306.

Anderson, J. R. (in press) Production systems, learning, and tutoring. In D. Klahr, P. Langley, and R. Neches (eds.), Self-modifying production systems: Models of Learning and Development. Cambridge, MA: Bradford Books/MIT Press.

Brody, M. J. (1985) Transfer of Knowledge Within the Lake Ontario Sport Fishery, Oceans 85, Marine Technology Society, Washington, DC.

Carey, S., Cognitive Science and Science Education, American Psychologist, vol. 41, no. 10, pp. 1123-1130.

Champagne, A.B. and Klopfer, L.E. (1984). Research in science education; the cognitive psychology perspective. In D. Holdzkom and PD Lutz (Eds) Research Within Reach: Science Education. Washington, DC: National Science Teachers Association.

Chi, M., Glasser, R., and Rees, E., (1982) Expertise in Problem Solving. In R. Sternberg (Ed), Advances in the Psychology of Human Intelligence, (vol. 1), Hilsdale, NJ: Lawrence Erlbaum.

Collins, A., and Loftus, E. (1975) A Spreading Activation Theory of Semantic Processing, Psychol. Rev. 82, 407-428.

Feigenbaum, E. A. (1983) Knowledge engineering: The applied side. In Intelligent Systems:The unprecedented opportunity, eds. J. E. Hayes and D. Michie, Ellis Horwood, Chicester, England.

Feigenbaum, E. A. (1988) The state of AI in medicine, Symposium on AI in Medicine, Stanford University, Palo Alto, CA.

Feifer, R.G., Dyer, M.G., and Baker, E.L. (1988). Learning Procedural and Declarative Knowledge. In Proceedings of the International Conference on Intelligent Tutoring Systems, University of Montreal: Montreal, Canada, pp. 499-505.

Hayes-Roth, B., and Hayes-Roth, F. (1978) Cognitive processes in planning. Rep. No. R-2366-ONR, Rand Corp., Santa Monica, Calif.

Hayes-Roth, B. (1980) Human planning processes. Rand Corp., Santa Monica, Calif.

Kopec, D. and Michie, D. (1983) Mismatch between machine representations and human concepts: Dangers and remedies. Report to the EEC, Subprogram FAST, Brussels, Belgium.

Newell, A. and Simon, H.A. (1972) Human Problem Solving. New Jersey: Prentice Hall.

Novak, J., and Gowin, D.B. (1984) Learning How to Learn. Cambridge University Press, Cambridge, England.

Resnick, L.B. (1983) Mathematics and Science Learning: A new conception. Science, 220, pp. 477-8.

Sleeman, D.J. and Brown, J.S. (eds., 1982) Intelligent Tutoring Systems. London: Academic Press.

LEARNING ARTIFICIAL INTELLIGENCE CONCEPTS: A TOOLKIT APPROACH

B. Lees
Dept. of Computing Science
Paisley College of Technology
Paisley, UK

A practically-oriented introduction to artificial intelligence (AI) is outlined, which forms part of a postgraduate course in information technology. The course accepts graduates in diverse disciplines, except computer science. However, the wide range of mathematical and programming skills of the students constrained the possible course content and teaching strategy. Prolog was chosen as the teaching language; but the AI systems to be studied were too extensive for the students to program completely, for themselves, in the limited practical time avilable. One solution is to provide for the students a set of higher level software tools, matching the systems to be studied. Previous research suggested that interactive model building may be a useful strategy in learning computing concepts; it was therefore decided to investigate such an approach in the AI course. Prolog was taught in parallel with the lectures. Several Prolog programs were also developed for the students to illustrate various artificial intelligence concepts and problem solving strategies. The programs formed a basis for students to experiment: to extend or amend to solve other similar problems, which were usually open-ended, to give scope for both the more proficient, and also the weaker programmers to tackle appropriate tasks. This approach appears to be very promising and has prompted further software developments. Present research activity is centred on the provision of a more extensive set of Prolog software tools which will enable students to construct models of a variety of AI systems.

APPRENTISSAGE DES NOTIONS DE L'INTELLIGENCE ARTICIFIELLE : APPROCHE BOÎTE À OUTILS

On donne ici les grandes lignes d'un cours pratique d'introduction à l'intelligence artificielle (IA), qui s'inscrit dans un pogramme d'études des deuxième et troisième cycles en informatique. Ce cours peut être suivi par des diplômés de diverses disciplines, sauf les diplômés en informatique. Étant donné les nombreux niveaux de connaissances en mathématiques et en programmation des étudiants, il a fallu restreindre le contenu du cours et rajuster nos stratégies d'enseignement. Le langage d'enseignement retenu est Prolog. Les systèmes d'IA à étudier étaient cependant trop longs pour que les étudiants puissent les programmer complètement durant le peu de temps accordé aux travaux pratiques. Une solution consistait donc à fournir aux étudiants un ensemble d'outils logiciels de haut niveau, adaptés aux systèmes à étudier. Des recherches antérieures avaient démontré que la construction d'un modèle interactif pouvait se révéler une excellente façon d'enseigner les notions d'informatique. Il a donc été décidé d'examiner la pertinence d'une telle approche pour un cours sur l'IA. Le langage Prolog a alors été enseigné parallèlement aux exposés du cours. En outre, on a rédigé plusieurs programmes en Prolog afin d'illustrer aux étudiants les diverses notions de l'intelligence artificielle et les stratégies de la résolution des problèmes. Ces programmes ont formé un corpus de départ à partir duquel les étudiants ont tenté des expériences : soit développer ces programmes, la plupart extensibles, soit les modifier pour résoudre des problèmes analogues. D'ailleurs, ils ont été profitables aussi bien aux programmateurs expérimentés qu'à ceux moins habiles. Cette approche semble très prometteuse et a été à l'origine de nouveaux développements de logiciels. Nos travaux actuels s'attachent surtout à constituer un ensemble d'outils logiciels en Prolog plus complet afin que les étudiants puissent construire les modèles de divers systèmes IA.

LEARNING ARTIFICIAL INTELLIGENCE CONCEPTS: A TOOLKIT APPROACH

Brian Lees

**Department of Computing Science,
Paisley College of Technology,
Paisley, PA1 2BE, Scotland.**

INTRODUCTION

Artificial Intelligence forms an important component of the computing science teaching programme at Paisley College. Course modules in artificial intelligence are currently offered to senior undergraduates in computing science, to postgraduate diploma students of information technology, and also to students on the postgraduate diploma course in computer education for teachers. In this paper the approach taken in teaching the postgraduate information technology course is discussed.

In an attempt to meet the demand for graduate entrants to the information technology industry who possess appropriate skills, postgraduate diploma courses in information technology have been developed in a number of institutions of Higher Education in the UK in recent years. Students accepted for the course at Paisley are graduates of diverse disciplines (except computing science). After completing the core modules of the course, which include various computing, engineering and business studies subjects, students select an option, of which one is a module in intelligent systems. This module runs for 4 hours per week, over 10 consecutive weeks.

A problem which is faced at the outset in teaching the intelligent systems module is the wide range of mathematical and programming abilities, and backgrounds, of the students, which therefore constrains the possible course content and teaching strategy.

COURSE PHILOSOPHY

The broad aims of the intelligent systems course are to provide students with an introduction to:

- the fundamental nature and concepts of artificial intelligence,
- programming in an AI language,
- the areas of application of AI.

The subject of artificial intelligence encompasses a wide range of applications, and demands an understanding of concepts derived from various disciplines; for example, computer science, psychology, linguistics, formal logic.

AI may be viewed as either a scientific or engineering activity. In the scientific approach, the main area of concern is the understanding of the mechanisms underlying human intelligence. Here, the computer is used as a simulation tool with which to test hypotheses about intelligence. In contrast to this, the aim in the engineering approach is to construct, in the computer, a system having the capacity to exhibit some aspect of intelligent behaviour. In teaching a course in AI from a computing science perspective, it is the latter of these approaches which is the more appropriate, and is the approach which was followed.

A problem faced at the outset was the rather limited teaching time available. It was decided to concentrate largely on the central concepts and methodology of the subject, backed up by a study of a few carefully chosen application areas. It was thus envisaged that those students who wished to continue the study of AI would then be in a favourable position to gain a broader coverage of the subject by further reading and self study.

The areas covered in the course include:

- the characteristics of intelligent systems, particularly humans,
- the features of problems that render them appropriate for an artificial intelligence approach,
- symbolic processing,
- the principles of AI programming,
- problems expressed as search through a search space,
- problem solving methods,

- knowledge representation,
- principles of natural language processing,
- the application of the above to expert systems.

POSSIBLE TEACHING APPROACHES

Because the limited mathematical knowledge of many of the students precluded a highly analytical approach, one possibility would have been to have taught a purely theoretical course, including, perhaps, student seminars in which pertinent issues could be discussed. Such a course was considered to be unsatisfactory, since there was a danger of the course being too superficial.

It was intended to teach a practically-oriented course. One way to achieve this would have been to teach the essentials of an AI programming language, supported by a set of relevant programming assignments. But in the limited time available the students would probably not have been able to proceed much beyond simple text book programming exercises; the complexity of many AI systems would have precluded their inclusion in the assignments. There was also the strong possibility that the result would have been merely a programming course, which was certainly not the intention.

Another approach would be to use one of the commercially available expert systems shells as a basis for practical work. This can be a relatively speedy way to construct certain types of expert system. But the application needs to be appropriate to the structure of the shell. This has some merit in a course restricted to expert systems alone, but the course aimed to cover wider issues than this. It would, in any case, have presented a user view of expert systems and would have given the students little, if any, insight into the architecture and mechanisms underlying the operation of such systems.

A STRATEGY FOR LEARNING AI CONCEPTS

The discussion so far suggests an approach in which the students have adequate practical activities. However, it is not really necessary for them to program everything for themselves. An alternative strategy would be for the students to build models of relevant systems. For this to be achieved, the students need to be provided with a set of appropriate tools. By adopting this approach it was also hoped to overcome the problem of the limited programming expertise of some of the students.

This is a philosophy which also underlies a number of other applications of artificial intelligence to education. For example, in research with school pupils, Howe et. al. provided pupils with a collection of modelling tools using the Logo language, with the aim of assisting their understanding of mathematical concepts[1]. This idea was also used by Goodyear in teaching a course in Environmental Systems[2]. A similar approach has been adopted by the author in teaching concepts in other branches of computing science. An integrated modelling environment was constructed to enable students of computer technology to build models of computer architecture structures - at both software and hardware levels. This was found to be a useful way of assisting students' understanding of concepts of computer architecture[3]. Also, in teaching a systems-oriented course in concurrent processing, students were provided with a set of concurrency primitives, written in the language Modula-2, to enable them to model and test various mechanisms for the synchronisation and communication of concurrent processes[4].

THE COURSE

After careful consideration, it was decided to proceed on two fronts simultaneously; theoretical lectures were given in class, which were supported by a practical introduction to AI programming. For the first few weeks these activities proceeded independently, to enable the basic nature and scope of AI to be discussed, and also for the students to become familiar with the basic concepts of AI programming (guided by laboratory worksheets).

Two key themes central to the course are the concepts of (i) knowledge-based programming, and (ii) goal directed search, as problem solving methods. These ideas pervade all aspects of the course.

The Programming Language

For a programming language there are several possibilities. Lisp could have been used. As many of the classical programs in AI have been written in Lisp

this would have had the advantage of enabling the students to study some of these systems. Such an approach would be a viable possibility for the senior computing science students, but it may be argued that Lisp is perhaps not the most readable of languages for use by students. It was thought also that the overhead of teaching the students to program in Lisp would be too great. Another possible language is POP11[5], but this was rejected because it is less universally known than its competitors.

The programming language chosen was Prolog, for various reasons. It was believed to be more suitable for use by the information technology students, with their varied programming experience and abilities, because of its higher level declarative logic programming style of programming, which can result in rather powerful, but short programs. Prolog has also been found to be appropriate elsewhere for teaching AI programming to non-scientists[6]. It was believed that Prolog would allow a fresh start to programming, particularly for those students who had experienced some difficulties with programming in a purely procedural style in earlier modules of the course. Indeed, it has been suggested that learning to program in a declarative manner can be easier for those without too much exposure to the traditional procedural style. From the author's experience of teaching programming, any languages or tools which can ease the burden of students from the necessity of investing many hours in constructing vast quantities of detailed code is certainly worth trying.

The aim was to teach a problem-centred, rather than a language-centred course. It was certainly not intended to teach all the facilities available in Prolog. Also there was insufficient time to embark on an extensive treatment of the theory of logic programming. Clearly some compromises needed to be made, by adopting a less than rigorous coverage of some of the material, in the interests of acquainting the students with a variety of ideas and applications.

Sufficient Prolog was taught to enable the students (i) to write simple list processing and recursive procedures, (ii) to enable them to understand the structure and operation of larger programs, (iii) to enable them to adapt pre-written programs to meet the needs of different problems, and (iv) to construct similar programs from the building blocks provided.

Students are encouraged to develop short procedures, and to test these thoroughly, in a bottom-up fashion, before using them in other modules. Some of the students had difficulty in constructing recursive procedures. For students without a strong mathematical or science background, recursion can be a conceptually difficult topic. With the aim of helping them to overcome this difficulty, the students were encouraged, wherever possible, to write down a specification of the required recursive procedure, before expressing it in Prolog. This did help to a certain extent.

The programming environment used is a laboratory containing a local area network of Vaxstation workstations supporting Ultrix. This was convenient since it gave students easy access across the network to the software that had been prepared for them.

Course Materials

Various example Prolog programs were developed for the students, and were discussed at appropriate points in the course. These programs formed a basis for experimentation, and also for modification, to solve other similar problems. These problems were ofter open-ended and allowed the weaker students to attempt rather modest goals, while those with greater programming proficiency had scope to tackle more ambitious tasks.

Programs provided for students include: simple knowledge-based programs for the solution of puzzles, programs illustrating the use of simple heuristics, general problem solver, simple natural language programs depending on key-word recognition, parsers for simple context-free languages, and for Prolog programs, All these form a basis for students to experiment or to adapt to suit the needs of other problems. The students are guided in these practical activities by accompanying laboratory worksheets.

A set of general purpose software tools was also provided, to be used as building blocks in constructing student programs. These modules include: a collection of list processing operations, primitives for input-output and for the display of structured information (e.g. a parse tree), a lexical analyser for use as a pre-processor to natural language

programs, and primitives to trace program execution.

DISCUSSION

A very similar approach to teaching AI has also been used with final year students of computing science. However, in contrast to the information technology students, these students have a more uniform computing background and are already competent programmers in other branches of computing. This background enables more substantial programming assignments to be tackled, though, as with the information technology students, the provision of much pre-written material enables more to be accomplished.

One of the main problems encountered, as had been previously anticipated, was how to reconcile the wide differences in programming ability and experience of the students. The course is run as an option and this helps to some extent, since not all students are forced to do the course. Students from a wide range of backgrounds - from forestry, linguistics, the physical and human sciences, to various forms of engineering - have successfully completed the course. Feedback from students at the end of the course has been very positive, and has prompted further work on the development of the software tools. Work continues with the aim of developing and testing with students a more extensive set of modelling primitives, in an integrated modelling environment.

The experience of using Prolog has, in the main, been satisfactory.A bonus of using Prolog is its relatively simple syntax and absence of keywords and structured statements. But there are inevitable problems. In the early stages of logic programming, which simply involves database construction and interrogation students have little difficulty. However, a major problem is faced by some students in understanding the concepts involved in recursive programming. Here, students with a mathematical, but not necessarily computing, background have an advantage; they may see the comparison with recurrence relations and with mathematical induction. Even with the provision of modelling tools there is still a need to 'cement' the building blocks together, and this almost always involves the writing of recursive procedures. Quite how this obstacle is best surmounted is a topic for on-going and future research[7]. There is a need for an AI programming environment designed to assist the novice programmer in program construction; this forms the basis of a further proposed research project.

REFERENCES

1. HOWE J.A.M. Model building, mathematics and LOGO. In: New Horizons in Educational Computing (Edited by M. Yazdani), Ellis Horwood, 1984.

2. GOODYEAR P. A toolkit approach to computer-aided systems modelling. 6th Computers in Higher Education Conference, University of Lancaster, April 1986.

3. LEES B. Learning Computer Architecture Concepts through Interactive Model Building. Ph.D. Thesis, CNAA/Edinburgh, 1987.

4. LEES B. Introductory concurrent programming with Modula-2. ACM SIGCSE Bulletin, vol 17(3), pp 34-41, 1985.

5. BARRETT R., RAMSAY A. and SLOMAN A. POP11: A Practical Language for Artificial Intelligence. Ellis Horwood, 1985.

6. BUNDY A. Teaching AI programming to non-scientists. Procs. AISB International Conference on Artificial Intelligence and Education, Exeter University, Sept. 1985, pp 16-19. 1985.

7. WIEDENBECK S. Learning iteration and recursion from examples. Int. J. Man-Machine Studies, vol 30, pp 1-22. 1989.

COMPUTERIZED DIDACTIC LABORATORY

G. Cappello, C. Gabbianelli, M. De Pinto, A. Arullani
IX Semeiotica Chirurgica
University of Rome - La Sapienza
Rome, Italy

A computerized laboratory can support University courses. MacTEST is a Macintosh program for the management of computerized multiple-choice questions. Computerized tests can help the student in the evaluation of his proficiency: during the time of preparation of the exam the student freely and autonomously undergoes regular examinations to test his preparation. The computer can analyze the most significant gaps and write down a detailed report for the student only. Knowing which topics must be considered as acquired and which must be reviewed, the length of preparation will be shortened. The general structure of the test supposes that the basal knowledge to be tested is arranged in more levels: lower levels contain the fundamental concepts while the higher ones contain more complex notions which can sometimes be derived from a lower level topic. The computer tests, at the beginning, the highest level, and, if the answer is wrong, it goes down to lower levels to evaluate how deep is the gap of the preparation of the student. As a matter of fact, each student will be submitted to a different number of questions depending on his preparation. Using an APPLE MACINTOSH computer the mouse helps the student in making his choices: it takes completely the place of the keyboard.

LABORATOIRE DIDACTIQUE INFORMATISÉ

Un laboratoire informatisé peut servir de complément à des cours universitaires. MacTEST est un programme sur Macintosh qui gère des questions à choix de réponse informatisées. Des tests informatisés peuvent aider un étudiant à évaluer sa maîtrise du sujet : durant sa préparation à un examen, l'étudiant se soumet librement à des tests réguliers pour apprécier sa préparation. L'ordinateur analyse les lacunes les plus importantes et rédige pour l'étudiant seulement un rapport détaillé. L'étudiant connaît ainsi les aspects qu'il maîtrise et ceux qu'il doit encore travailler, ce qui écourte la période de préparation. La structure générale du test suppose que les connaissances de base à éprouver comportent plusieurs niveaux : les niveaux inférieurs contiennent les notions fondamentales, tandis que dans les niveaux supérieurs on retrouve les notions plus complexes parfois dérivées d'un sujet de niveau inférieur. L'ordinateur évalue, au départ, le niveau supérieur et, en cas de mauvaise réponse, passe au niveau inférieur; ainsi, de niveau en niveau, il peut apprécier le manque de préparation de l'étudiant. En fait, chaque étudiant est soumis à un nombre différent de questions selon l'état de sa préparation. Avec un APPLE MACINTOSH, la souris permet à l'étudiant de faire ses choix; en fait, cette souris se substitue complètement au clavier.

COMPUTERIZED DIDACTIC LABORATORY

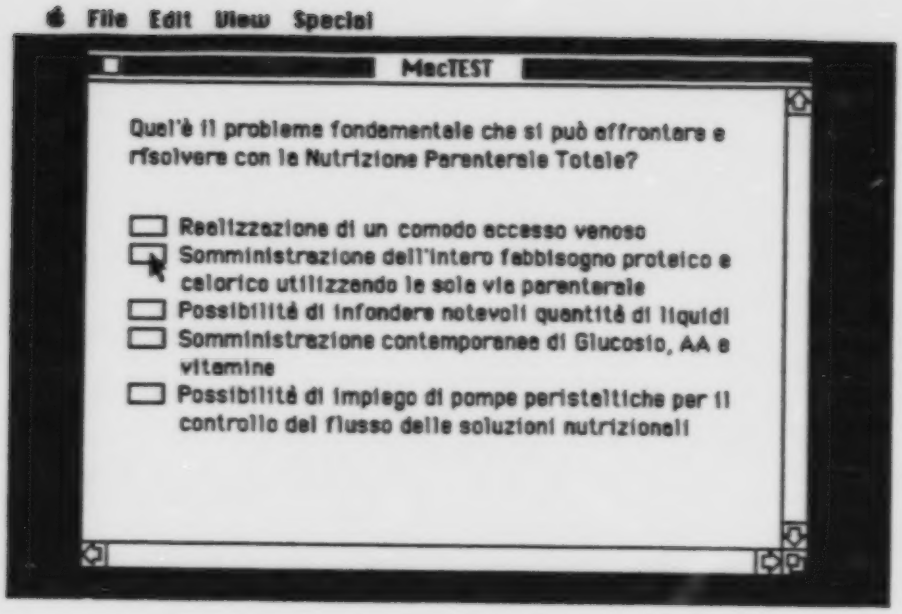
G.Cappello, C.Gabbianelli, M.De Pinto and A.Arullani

IX Semeiotica Chirurgica,
University of Rome - La Sapienza -,
Rome, Italy

Personal computers are used extensively in professional work: they are useful in medical treatment and diagnosis. Actually there is an increasing trend to use them in medical teaching. As a matter of fact personal computers can be used to support medical courses: they cannot take the place of a professor, but they can assist medical education in many ways. Computerized simulation, for instance, is a very effective educational tool: simulations are possible in many medical and biological fields and they can be very stimulating.
Also multiple choice questions can be used on a personal computer support: they are ideally suited to computer methods.
Aim of this study is the development of a multiple choice questions computer program for the development of a university course in Surgical Pathology.

METHODS

A multiple choice question program has been developed on an APPLE Macintosh Plus: we chose this type of personal computer because it is provided with the "mouse" which is a very helpful appendix of the computer: you can easily move an index on the screen to activate buttons and scroll bars reducing the need of the keyboard. Macintosh, also, for its graphic facilities, allows a very elegant and tidy composition of program screens. Computers are grouped in a laboratory which is freely open to the students of the course.
The multiple choice question program is designed for helping each student during the preparation of the exam of Surgical Pathology. The student can choose the topics in which he wants to test his performance. The computer will propose a series of multiple choice questions: at the end of the test the computer will supply the student with a general evaluation of his proficiency and a list of topics which must be reviewed. No record will be left of the computer evaluation so that the student can be sure that it will not influence the final exam.
Questions can have one or more possible solutions, depending on the topic which is concerned.
The general structure of the test supposes that the basal knowledge to be tested is arranged in more levels: lower levels contain the fundamental concepts while the higher ones contain more complex notions which sometimes can be derived from a lower level topic.
The knowledge is divided in single trees branched in a variable number of topics: for each branch the computer tests, at the beginning, the highest level, and, if the answer is wrong, it goes down to lower levels to evaluate how deep is the gap of the preparation of the student. If the answer is correct it skips to next branch and then next tree till the end of the exam. As a matter of fact each student will be submitted to a different number of questions depending on his preparation.
The program has three main functions:
- it allows the storage of the whole set of questions.
- it controls the flow of questions on the base of the student's answers.
- it allows the final evaluation of the test, it gives an opinion on the student's preparation and it print out a list of topics which must be reviewed.



RESULTS

The program is very simple: it has been written and compiled with Microsoft BASIC V.3.0.
The core of the program is the question database which has been written with a word processor and saved as a text file: each question is catalogued and for each possible answer the mark to be assigned and a pointer to next question are specified.
The mouse helps the student in making his choices: it takes completely the place of the keyboard. We think this is a main point in our experimentation: writing on a keyboard is an unusual habit for most medical students while moving a mouse is easy and natural as moving a finger or a pen.
A limited experimentation of the program has been done on medical student for a total of 325 tests. From this experience some conclusions can be drawn:
- the system is easy to use and extremely user friendly: the student learns immediately how to work with the mouse and the shift of questions is practically immediate;
- generally students accept the computer as an examiner: they like this new approach to medical education because it is anonymous and it cannot compromise the result of the final exam;
- the printed list of topics which must be reviewed is very useful for the student: is the program of the study to be done. It can be improved when we will get more experience on this computerized approach. The immediacy of the response makes the computer an attractive learning aid.

COMMENT

Not computerized multiple choice question exams are not uncommon in our university: the student is supplied of a small booklet of questions and he must indicate his answers by placing a mark with a pencil. The evaluation of the test can be done by the student himself or by the teacher.
The difference with the computerized approach is that the former is technically much faster in the collection as well as in the evaluation of re-

sults. Moreover the computer on the base of each answer of the student can decide if a number of questions, on a topic that is probably well known by the candidate, can be skipped, further on reducing the length of the test.
There has been much discussion about the desirability of multiple choice questions: it has been said that they concentrate on small details rather than testing a candidate's ability to reason: this depend, of course, on how well the questions are devised. It depends also on the kind of knowledge to be tested: we think that Surgical Pathology can be easily adapted to a multiple choice examination.
The mouse makes the program more user-friendly. This is a very important point because multiple choice questions tend to be arid and tedious: we must make any attempt to improve its acceptation from the student. The mouse is a great help to make the testing fast and easy, handy and adaptable.
We think that this 'personal' multiple choice examination can have a great future and it can be developed with the utilization of compact disks which can connect the program with an enormous data-base of pictures and schemes.
Computerization of professional work is already widespread, but this is only a beginning: it must be enhanced. The use of computer in university courses can be for the student a start with computers, the first introduction of a very effective tool in a professional life.

REFERENCES

Arullani A., Cappello G., Capuano C., Venosi S., Casciani C.U., And Ricci C.
Surgical treatment of esophageal stenoses: a computerized approach for nutritional esophageal disorders. In: T.R. De Meester, Raven Press, New York,1985.

Arullani A., Cappello G., Venosi S.: Automatic management of a total parenteral nutrition unit with a personal computer. Proceedings of 6th Congress of ESPEN, Milan, 1984

Arullani A., Cappello G., Venosi S., Borghese L., Ricci C.: A computerized program for nutritional control and treatment. Proceedings of 8th World Congress of CICD, Amsterdam, 1984

Arullani A., Cappello G., Venosi S., Borghese L. Informatica ed alimentazione. Proceedings of "Alimentazione Enterale e Parenterale in Chirurgia Digestiva", Napoli, April 18,1984

Arullani A., Cappello G., Meschini L., De Vittori R, Casciani C.: Automatic management of patients on TPN with a computer program. Clinical Nutrition 1:67,1982

Arullani A.: Computer e Chirurgia. Proceedings of Accademia Lancisiana, Roma,1986

UN CONSTRUCTEUR D'ÉNONCÉS DE PROBLÈMES DE PHYSIQUE

H. Giroire-Brousse
Laforia
Université Pierre et Marie Curie
Paris, France

Les systèmes d'EAO utilisent classiquement une batterie de problèmes préenregistrés. L'introduction des techniques de l'IA conduit naturellement à vouloir créer des exercices à la demande du Module Tutoriel, en fonction des aptitudes de l'élève. Dans un Tuteur Intelligent cette création relève du Module Expert. C'est lui en effet qui a la connaissance du domaine et la capacité de résolution nécessaires. Par contre, le contrôle de cette activité relève du module tutoriel.

C'est cette tâche de création automatique de problèmes, génération de l'énoncé en langue naturelle comprise, qui est l'objet du travail que nous présentons ici. A notre connaissance, elle n'a pas été abordée de manière systématique. Nous nous sommes placée dans le cadre de l'enseignement de la physique tel qu'il est fait dans les lycées français. Notre Constructeur d'Énoncés de Problèmes (CEP) est un Système Expert écrit en SNARK (LAURIERE 85), système expert essentiel d'ordre 2. Pour réaliser CEP, nous avons dégagé, de l'étude des exercices de physique, des stratégies de construction de trames de problèmes: stratégies de 'substitution de grandeurs attachées à des objets formels', de 'duplication d'expérience', .. Nous décrivons ces stratégies qui permettent à CEP de construire une trame formelle qu'il plonge ensuite dans le monde réel en habillant les objets formels (réci- pient, résistor, ..) par des objets réels (bidon, filament de lampe, ..) et en valuant les grandeurs associées à ces objets. Après une étude linguistique des exercices des manuels, nous montrons alors comment CEP engendre un énoncé en français exprimant la trame habillée. Nous terminons par quelques exemples d'exercices construits par CEP.

A CONSTRUCTOR OF EXPRESSIONS OF PHYSICS PROBLEMS

CAL systems traditionally use a set of prerecorded problems. The introduction of AI techniques naturally leads to the desire to create the exercises demanded by the tutoring module, as a function of pupil skills. In an intelligent tutor, this creation falls to the expert module, which in fact contains the required knowledge of the field and necessary solution capability. On the other hand, control of this activity is provided by the tutoring module.

This task of automatic problem creation, with generation of expressions in understandable natural language, is the subject of the study I will be presenting. To my knowledge, it has not been examined systematically before. My context has been the teaching of physics as it is done in the lycées of France. Our problem expression constructor (CEP) is an expert system written in SNARK (LAURIERE 85), an essential order 2 expert system. To create CEP, we examined physics exercises, from which we derived strategies for constructing problem frames; these strategies involved "substitution of sizes attached to formal objects", "duplication of experience" and so on. I will describe those strategies that allow CEP to construct a formal frame which it then plunges into the real world, actualizing the formal objects ("container", "resistor" and the like) with real objects ("can", "lamp filament" and so on) and providing values for sizes associated with these objects. After a linguistic study of the exercises in textbooks, I will show how CEP generates a statement in French expressing the frame that has been actualized. I will conclude with a few examples of exercises constructed by CEP.

UN CONSTRUCTEUR D'ENONCES DE PROBLEMES DE PHYSIQUE

Hélène GIROIRE-BROUSSE

Université Pierre & Marie Curie - LAFORIA
Tour 45-46 - 4, place Jussieu - 75252 PARIS CEDEX 05 - FRANCE

INTRODUCTION

L'utilisation de l'ordinateur dans l'enseignement se développe et se développera dans les domaines les plus divers. L'ordinateur permet en effet d'une part de soulager l'enseignant des tâches répétitives et d'autre part de mettre à la disposition de l'élève un maître, moins subtil sans doute, mais totalement disponible et infiniment patient. Les premiers systèmes ont été construits dans le cadre de l'Enseignement Assisté par Ordinateur (EAO). Leur rigidité a incité les chercheurs à adopter une approche Intelligence Artificielle (IA). Ils ont ainsi construit, à l'aide des techniques d'IA, des systèmes dits d'Enseignement Intelligemment Assisté par Ordinateur (EIAO). Leurs principaux composants sont en général :

- un module expert contenant l'expertise du domaine,
- un module tuteur (pédagogue, tutoriel),
- un module interface en langage naturel ou utilisant les derniers outils informatiques de communication,
- un modèle de l'élève.

Nous renvoyons à [NIC 88] pour une synthèse des réalisations et tendances de recherche dans ce domaine ainsi qu'à [WEN 87] pour une présentation détaillée.
Tout enseignement utilise des exercices et problèmes. Ils servent :

- à faire assimiler par l'élève les concepts nouveaux,
- à lui apprendre à les manipuler, éventuellement jusqu'à ce que cela devienne un automatisme (la patience du maître est alors fort utile),
- à permettre au maître de vérifier que cet apprentissage est bien fait.

Les systèmes d'EAO utilisent classiquement une batterie de problèmes préenregistrés. L'introduction de l'IA conduit naturellement à vouloir créer des exercices à la demande du maître ou du module tutoriel, en fonction des aptitudes que manifeste, ou au contraire ne manifeste pas, l'élève (c'est là qu'apparaît l'intérêt de la capacité du système de suivre individuellement, le temps qu'il faut, chaque élève).
Cette création relève du module expert. C'est lui en effet qui a la connaissance du domaine et la capacité de résolution nécessaires. Par contre, le contrôle de cette activité relève du maître et/ou du module tutoriel.
C'est cette tâche de création automatique de problèmes, y compris la génération de l'énoncé en langue naturelle, qui est l'objet du travail que nous présentons ici. A notre connaissance, elle n'a pas été abordée de manière systématique. C'est évidemment un sujet très vaste comportant de nombreux aspects. Pour pouvoir l'entreprendre, il nous a fallu nous placer dans un cadre suffisamment riche, pour que notre étude soit significative et relève vraiment de l'IA, tout en étant suffisamment restreint pour pouvoir parvenir à des résultats. Mieux encore, nous avons choisi un cadre à "géométrie variable". Nous voulions qu'il puisse être facilement restreint, s'il se révélait trop ambitieux à un certain stade de développement de l'étude. Nous voulions aussi qu'il puisse être commodément étendu, lorsque les capacités de notre système le permettraient.
Définir de tels domaines, formant un tout cohérent et clos, c'est-à-dire dont les relations avec le milieu extérieur peuvent être oubliées, est une affaire de spécialistes. Elle ne peut être entreprise que par des experts, et ce n'était pas l'objet de notre travail de procéder à une telle construction. C'est pourquoi nous avons choisi des domaines déjà constitués. Nous nous sommes placée dans un cadre de problèmes élémentaires du monde réel (champs, récipients,...) et de la physique (courant continu, calorimétrie,...).
Dans le domaine choisi comme objet d'un enseignement, il faut savoir quels sont les objets du domaine, les concepts mis en jeu, quelles sont les relations (qualitatives et quantitatives) entre eux. Il faut connaître le lien entre ces nouveaux concepts et les connaissances antérieures de l'élève. Il faut définir un ordre d'apprentissage, prenant en compte le niveau de difficulté, les erreurs et confusions les plus fréquentes, ainsi que celles faites par chaque élève,... C'est le rôle des didacticiens de la discipline de faire cette étude et d'en déduire des stratégies de présentation aux élèves du nouveau domaine. Ces stratégies incluront, en particulier, une politique d'utilisation d'exercices et problèmes.
Nous avons donc conçu et réalisé un système nommé CEP (Constructeur d'Enoncés de Problèmes), de génération automatique de problèmes, en amont duquel nous supposerons l'existence d'un module matérialisant le travail des didacticiens et guidant la construction de nos énoncés.
Les problèmes et les exercices rencontrés dans les manuels usuels de physique ont été l'objet de critiques de la part des didacticiens de la discipline. Ces derniers soulignent en particulier le caractère souvent artificiel de ces énoncés.
La critique la plus grave est que l'activité la plus importante du point de vue de la physique, la modélisation, est contrainte dans les énoncés de ce type qui demandent en fait à l'élève de reconnaître le modèle à utiliser pour répondre aux questions.
Cependant les diverses critiques n'empêchent pas les didacticiens de reconnaître l'intérêt des exercices classiques (*coutumiers* [DUM 87]). Ces derniers permettent aux élèves de se familiariser avec les concepts nouveaux, de se les approprier en les manipulant, de créer des automatismes. Les exercices générés par CEP sont donc du type coutumier.

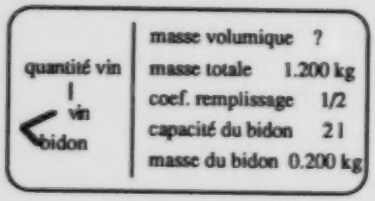
MÉTHODE DE CONSTRUCTION DÉGAGÉE

Tout exercice ou problème dans un domaine du monde réel décrit une portion d'univers, constituée d'un ou plusieurs objets reliés entre eux, et dont les liens avec le reste de l'univers sont occultés. Dans les exercices que nous avons envisagés, à chaque objet sont attachées des grandeurs ayant des valeurs reliées par des équations. On peut aussi considérer que chaque exercice décrit une expérience dont on demande à l'élève de prévoir le résultat. La détermination de

la valeur d'une des grandeurs est la question pour l'élève. Cette description, même abstraction faite de la question, n'est jamais complète. Elle fait référence, en général implicitement, à de nombreuses connaissances que l'élève est supposé avoir. Une des principales causes d'échec de l'apprentissage est d'ailleurs la référence à des connaissances en fait non ou mal acquises, et le diagnostic à faire dans ces situations est du ressort du module tutoriel s'appuyant sur son modèle de l'élève, mais ceci sort du cadre de CEP.
Analysons un exemple créé par CEP de tels exercices :

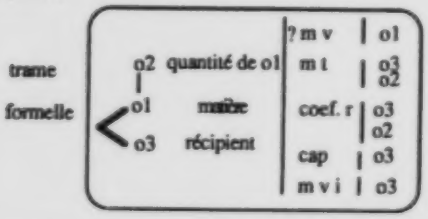
Un bidon vide de capacité 2l pèse 0.200 kg. La masse du bidon rempli à moitié de vin est 1.200 kg. Quelle est la masse volumique du vin?

Nous pouvons distinguer deux aspects, d'une part un texte en français, d'autre part un contenu de l'exercice, contenu que nous appelons trame. Cette trame repère les grandeurs et les objets nécessaires à l'élaboration de l'exercice et les liens qui les sous-tendent. Pour l'exemple ci-dessus la trame est:



Le texte n'est que l'expression en français de cette trame et son élaboration ne peut être que la dernière étape du processus de création.

La trame comporte elle-même deux aspects, l'un pragmatique, l'autre formel.



La trame est obtenue en "habillant" la trame formelle c'est-à-dire d'une part en faisant correspondre aux deux objets formels récipient et liquide deux objets réels bidon et vin compatibles entre eux, d'autre part en attribuant des valeurs plausibles aux grandeurs. Cet habillage constitue naturellement la deuxième étape pour CEP du processus de création d'exercices.
La trame formelle n'est que la partie visible d'une modélisation complète de l'expérience proposée à l'élève.Sa construction peut être abordée de façons diverses. Nous avons pu constater qu'un grand nombre de problèmes de physique de l'enseignement concernant le début d'une théorie sont bâtis autour de formules, ou du moins peuvent être recréés ainsi. Ces formules constituent une modélisation d'expérience, parfois très élémentaire. En combinant plusieurs d'entre elles, nous arrivons à décrire d'autres expériences plus complexes.
Les exercices proposés dans les recueils ayant souvent pour but de faire assimiler les grandeurs ou formules introduites nous avons fait démarrer le processus de création d'exercice par une grandeur ou une formule sur laquelle on désire faire travailler l'élève et un niveau de difficulté souhaité. Actuellement ce sont des données que doit fournir l'utilisateur (l'enseignant), lorsqu'il désire que CEP lui construise un exercice. Comme nous l'avons dit plus haut, dans une perspective d'intégration de CEP dans un système d'EIAO, ces données seraient choisies par le module tutoriel.
CEP, à partir de ces données, construit donc un exercice en trois étapes.

CONSTRUCTION DE LA TRAME FORMELLE

CEP bâtit donc tout d'abord une *trame formelle* faisant intervenir des grandeurs (volume, masse volumique, ddp,...) relatives à des objets formels, c'est-à-dire en fait à des classes d'objets (récipient, matière, résistor,...) [Giroire-Brousse H. 1987]. Le processus de création de trame démarre d'une grandeur ou d'une formule sur laquelle on veut faire travailler l'élève. CEP a dans ses connaissances les formules exposées dans le cours, elles modélisent les expériences élémentaires (par rapport au cours). Une formule comprend les grandeurs, les objets auxquelles elles sont attachées, les liens entre ces objets et enfin l'équation reliant entre elles les valeurs de ces grandeurs. Nous avons dégagé plusieurs stratégies générales (indépendantes du domaine). La stratégie de *substitution*, les stratégies de *duplication*, l'une *autour d'un objet*, l'autre *autour de la valeur d'une grandeur*. La stratégie de substitution consiste à utiliser une formule pour substituer, à une grandeur déjà présente dans la trame partiellement construite, les grandeurs permettant de l'exprimer à l'aide de la formule correspondante. Dans l'exemple présenté au-dessus, CEP, à partir du concept *masse volumique* donné, a selectionné la formule qui définit celle-ci puis il a substitué à la masse du liquide le coefficient de remplissage d'un récipient et la capacité de ce dernier. Puis CEP a substitué à la masse du liquide le poids total récipient-liquide et la masse du récipient-vide. CEP prend des instances des différentes fomules, il est amené à identifier certains des objets qui interviennent en respectant les usages (ici on utilise le même récipient pour déterminer le volume et la masse du liquide). La stratégie de duplication autour d'un objet correspond à une démarche d'étalonnage, on fait deux expériences similaires en utilisant un objet commun, une de ses grandeurs caractéristiques sert de lien entre les deux expériences et permet de construire une trame. La duplication autour d'une valeur de grandeur revient à utiliser une égalité arithmétique entre les valeurs de deux grandeurs associées à des objets différents, cette dernière stratégie peut être utilisée lorsque la stratégie autour d'un objet n'est pas possible.

HABILLAGE DE LA TRAME FORMELLE : TRAME REELLE

Ensuite, CEP *habille* cette trame en une *trame réelle*, en choisissant, pour chaque objet formel, un objet réel représentant de sa classe (fût, eau, filament de tungstène,...). Puis, CEP attribue à chaque grandeur de la trame une valeur. Tous ces choix sont interdépendants et fortement contraints. Le choix du cuivre impose la valeur de sa chaleur massique, celui du filament d'ampoule implique le tungstène comme constituant, ce qui fixe la valeur de la masse volumique et de la résistivité. Le choix de certains objets réels peut entrainer la suppression d'une grandeur de la trame si sa valeur doit être connue de l'élève (par exemple : la masse volumique de l'eau). Les domaines de valeurs pour les diverses grandeurs sont de natures différentes et dépendent des objets réels pour lesquels ces grandeurs sont définies. Nous avons adopté comme heuristique de valuer les grandeurs attachées aux objets réels dans un ordre allant des grandeurs les plus contraintes aux moins contraintes afin d'éviter de se retrouver dans une impasse. Ceci est une heuristique de parcours d'un

espace de recherche, une solution étant un jeu de valeurs pour les grandeurs du contexte, qui satisfait les contraintes.
C'est pourquoi nous valuons dans l'ordre :

- les grandeurs de type rapport,
- les grandeurs dont le domaine de valeurs attachées à un objet réel se réduit à un point,
- les grandeurs qui, attachées à un objet réel, ont un domaine de valeurs qui est un ensemble énuméré,
- les grandeurs qui, attachées à un objet réel ont un domaine de valeurs intervalle,
- les grandeurs qui, attachées à un objet réel, dépendent des précédentes par des contraintes de cohérence (largeur > longueur (rectangle), volume (liquide) < capacité (récipient)).

Les grandeurs restantes sont valuées grâce aux équations, à partir des valeurs attribuées antérieurement aux grandeurs primitives.

GÉNÉRATION DU TEXTE EN FRANÇAIS

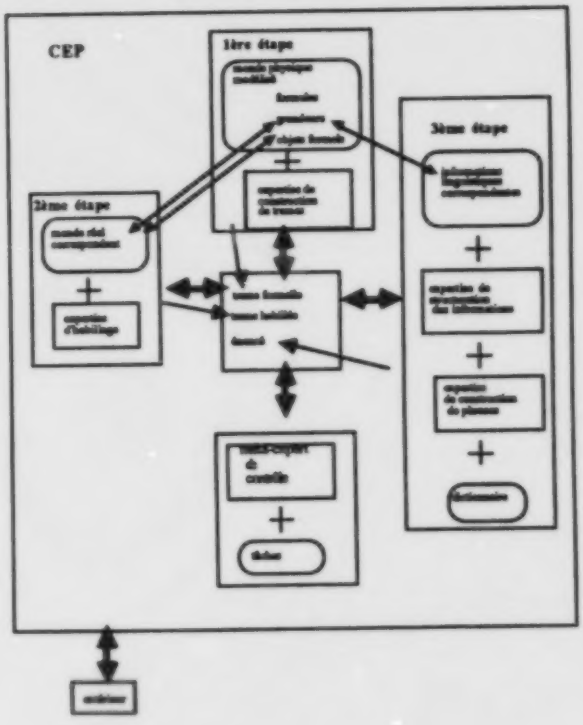
La trame réelle étant construite, il faut l'exprimer en français. C'est la dernière étape. Comme toute entreprise de génération de textes en langage naturel, celle-ci doit commencer par une étude linguistique du domaine, qui permette, en particulier, de mettre en évidence des mécanismes par lesquels le processus pourra démarrer. Le processus de génération d'énoncés de CEP est centré sur l'expression des grandeurs. Pour chaque grandeur CEP construit une phrase simple exprimant sa valeur. Ces phrases simples mises bout à bout ne composeraient évidemment pas un énoncé satisfaisant, même s'il exprimait toute l'information à transmettre à l'élève. CEP regroupe donc les phrases simples en phrases, plus ou moins complexes en fonction d'un niveau de regroupement demandé par l'utilisateur mais qui dans un système EIAO complet serait choisi lui aussi par le module tutoriel. CEP procède à ces regroupements en tenant compte des liens logiques entre les grandeurs exprimées. Il privilégie, par exemple, le regroupement de grandeurs associées au même objet. Cependant, les critères sémantiques ne sont pas les seuls à prendre en compte. Ils ne sont même pas toujours prépondérants. CEP procède ainsi à certains regoupements parce que les phrases simples correspondantes ont la même structure syntaxique ce qui permet des ellipses et donc une économie d'expression. En effet, comme le montre, en particulier, l'étude linguistique des manuels à laquelle nous nous sommes livrée, les auteurs humains sont très friands de tout ce qui permet une économie d'expression, et CEP, comme tout système automatique de génération, cherche à imiter le comportement de ces auteurs. Là encore, il y a de nombreux choix à effectuer, comme par exemple celui des formes des phrases simples. Nous montrons comment ces choix sont interdépendants et comment il est impossible de définir un ordre de priorité de ces choix, sans pouvoir exhiber aussitôt un ou plusieurs contre-exemples à la validité de ces priorités.

STRUCTURATION DES CONNAISSANCES

CEP procède donc en trois étapes. Chacune d'elles nécessite beaucoup de connaissances de natures diverses : connaissances scientifiques et pragmatiques pour concevoir un problème soluble et plausible, connaissances linguistiques pour composer l'énoncé en langage naturel. Il est de la plus grande importance de pouvoir exprimer toutes ces connaissances sous la forme la plus déclarative possible et de séparer les deux aspects, définition des univers d'une part, et stratégies de construction d'autre part. Il est également utile d'avoir une grande souplesse pour construire des objets nouveaux et pour pouvoir les compléter progressivement. C'est même indispensable dans un processus de création. L'exemple typique est celui de la trame de l'exercice, construite petit à petit au cours du processus. Ceci nous a orientée vers l'utilisation d'un système de production. Nous avons d'autre part été naturellement conduite à structurer la base de faits de notre système. Pouvoir accumuler des connaissances en vrac, ne veut pas dire, à notre avis, qu'il faille refuser d'introduire toute structuration, lorsque cela est possible. C'est donc ce que nous avons fait. Nous avons ainsi dégagé plusieurs notions de schémas. Un exemple typique est celui des schémas de formule, que nous avons constitués pour représenter toute l'information relative aux équations liées aux diverses grandeurs. Ce travail de structuration est ce que nous avons appelé *"utilisation d'un système de production dans une perspective orientée objets*. Pour ce faire, nous avons utilisé le système expert essentiel SNARK dans lequel nous avons introduit les notions de schémas qui nous étaient nécessaires. SNARK est un Système Expert Essentiel, avec variables, d'ordre 2, conçu par J.L. Laurière, modifié par lui-même et M. Vialatte [Laurière J.L. et Vialatte M. 1986].

Nous avons également structuré le travail de CEP en *tâches*. Comme nous l'avons déjà dit ce travail se décompose en trois étapes. Chacune de ces étapes se décompose en tâches : utilisation de la stratégie de substitution,..., choix d'objets réels, choix de valeurs,..., regroupements de phrases simples,..., écriture du texte,... Ces tâches font appel les unes aux autres de manière dynamique. Ainsi la tâche instanciation de schéma est-elle appelée par de nombreuses autres tâches. Chaque tâche est accomplie par un *agent* et ces agents doivent être coordonnés. C'est le rôle du *méta-expert* de faire ce contrôle de manière que le comportement de CEP, ensemble de tous ces agents, soit cohérent.

Voici l'architecture générale du système CEP



Notre système CEP travaille à ce jour, dans trois domaines : surfaces, récipients, résistors. Voici deux exemples d'exercices créés par CEP :

Un pré rectangulaire mesure 120 m de large et 134 m de long. Quelle est l'aire du pré?

créé à partir de la grandeur *surface*, et d'un niveau de difficulté *faible*.

La résistance d'un fil conducteur en cuivre de longueur 1.2m et de section droite 9 mm2 est 267 Ω. Quelle est la résistance d'un fil conducteur en cuivre de longueur 0.2m et de section droite 6 mm2?

créé à partir d'une part de la formule exprimant la résistance d'un fil en fonction de sa longueur, de sa section et de la résistivité du métal le constituant ; d'autre part d'un niveau de difficulté *moyenne* ; CEP a utilisé la stratégie de duplication autour de l'objet cuivre ; la résistivité a servi de grandeur pivot pour la construction du problème.

CEP comporte 313 règles, dont 165 pour le français, ainsi que de 2457 faits. Sur ces faits, 992 ont trait au français, 270 sont généraux, et 1195 sont liés aux domaines.
La base de faits comporte trois bases de connaissances, l'une regroupe les tâches au nombre de 34, données du méta-expert de contrôle, et les connaissances sur les attributs ; ces connaissances sont indépendantes du domaine physique traité, la base de connaissances linguistiques comporte d'une part les schémas de sorties (associés aux grandeurs) qui sont a priori utiles pour tout domaine et un dictionnaire partiellement lié au domaine, elle comporte un millier de faits, enfin la troisième base réunit les connaissances liées aux domaines physiques. Elle regroupe les formules, les grandeurs, les classes d'objets formels et les objets réels qui peuvent leur être associés.

CONCLUSION

Nous nous sommes intéressée à l'automatisation de la construction d'énoncés d'exercices de physique, tels que ceux que l'on rencontre dans les manuels des classes .
C'est pourquoi nous avons conçu et construit un système, CEP, de génération automatique de problèmes de type classique.
L'étude de manuels et recueils d'exercices nous a conduite à découper le processus de construction d'exercices en trois étapes successives : construction d'une trame formelle, habillage de cette trame, génération de son énoncé en français.
Nous avons ensuite dégagé, puis mis en œuvre, des stratégies de construction de trames (substitution, duplication,...). Ces stratégies sont indépendantes du domaine d'application considéré.
Cette indépendance du domaine se retrouve également dans les stratégies utilisées dans les deux étapes suivantes : habillage de la trame et génération de l'énoncé en français.
En ce qui concerne l'habillage, nous avons montré comment le choix d'objets réels et la valuation des grandeurs associées demandaient la satisfaction de contraintes multiples.
Nous avons aussi vu la difficulté de la génération de l'énoncé en français, difficulté qui s'exprime notamment par le nombre de faits et règles nécessaires (la moitié des bases de faits et règles de CEP), pour obtenir des textes simples.
Le passage par le langage naturel, est absolument indispensable pour des exercices de physique. En effet, ces exercices décrivent un monde informel, et une part essentielle du travail de l'élève consiste à passer de cette description informelle, à un modèle physique dont l'aspect quantitatif est décrit par un ensemble de formules. C'est la présence de cette activité de modélisation qui fait qu'il s'agit bien d'un exercice de physique. C'est elle qui nécessite donc la présence d'une interface importante en langage naturel, ainsi que de l'étape d'habillage de trame, qui n'est rien d'autre que la recherche d'un monde réel qui soit modélisable par le modèle que constitue la trame formelle.
Cette difficulté est spécifique : ainsi l'absence, en mathématiques, de ce travail de modélisation, fait que l'on ne rencontre dans les systèmes dédiés à ce domaine, ni le problème de l'habillage, ni celui du langage naturel.

Deux types d'utilisation de CEP par un enseignant peuvent être envisagés :
- soit pour générer des exercices sur les domaines que nous avons déjà prévus dans la base de faits actuelle,
- soit pour introduire un nouveau domaine physique.

L'enseignant devra alors faire l'analyse de ce domaine, puis exprimer le résultat de cette analyse, en termes de faits à rajouter dans la base de faits de CEP. Il devra également s'assurer que la base de règles de CEP convient encore pour ce nouveau domaine.

BIBLIOGRAPHIE

BRACHMAN R., SCHMOLZE J. 1985, *An Overview of the KL-ONE Knowledge Represeniation System*, Cognitive Science 9, pp171-216

DANLOS L.1985 *Génération automatique de textes en langues naturelles*, Masson, Paris.

DUMAS-CARRE A 1987, *La Résolution de problèmes en physique au lycée. Le procédural : apprentissage et évaluation* , Thèse de Doctorat d'Etat, Paris VII

GIROIRE-BROUSSE H. 1987, *D'une trame d'un problème à son énoncé en langage naturel*, Congrès international MARI/COGNITIVA 87, 18-22 mai, Paris, pp. 362-365.

GIROIRE-BROUSSE H. 1987, *Stratégies pour la Construction Automatique de Problèmes de Physique* , Colloque d'Intelligence Artificielle, Caen, publication n° 63 du LAFORIA, pp.79-120.

GIROIRE-BROUSSE H. 1989 *Réalisation d'un système expert dans une perspective orientée objets : une application en SNARK*. Convention IA 88-89, Paris, 23-27 janvier 1989.

GIROIRE-BROUSSE H. 1989*Un système à base de connaissances pour la génération d'exercices dans des domaines liés au monde réel*, Thèse de Doctorat de l'Université Paris 6, Janvier 1989.

JACOBS P. 1987, *Knowledge-Intensive Natural Language Generation*, Artificial Intelligence, v. 33, pp. 325-378.

LAURIERE J.L., VIALATTE M. 1986, *SNARK : a language for representation of declarative knowledge and an inference engine which uses heuristics*., IFIP Meeting, Dublin.

McKEOWN K.R. 1985, *Discourse Strategies for Generating Natural-Language Text*, Artificial Intelligence, vol. 27, pp.1-41.

NICAUD J.F., VIVET M. 1988, *Les tuteurs intelligents : réalisations et tendances de recherches*,T.S.I. n°1.

SEDLOCK D. 1988, *Natural Language Generation in a Large Expert System*. 8èmes journées internationales "Les systèmes experts & leurs applications", Avignon, pp. 401-411.

WENGER E. 1987, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmanninc., Los altos,California.

WINOGRAD T. 1983, *Language as a Cognitive Process*, vol.1: Syntax, Addison-Wesley Publishing Company.

MODEL ANALYSIS OF COMPUTER-ASSISTED TEST

J.-R. Wan, M.-J. Wu
Institute of Modern Educational Technology
East China Normal University
Shanghai, China

There are several various systems which must be tested frequently in our work. They may be a social system, a machine system, an educational system or a biological system. In this paper some problems about the model of CAT for an educational system will be discussed.

According to the special goals, the computer can give some input signals (which may be called "excitations") to the tested system and then the tested system feeds its output signals (which may be called "responses") back to the computer. To analyze the information coming from these responses by the computer system, we can decide fairly on the properties and functions about this tested system. Owing to the law of causation, the responses lag always behind their excitations by a certain time duration. We know the time duration can represent the speed of response of this tested system. Moreover, if these responses coincide with those expectations preset in the computer, then the tested system is correct, otherwise it is wrong. The results of test and analysis given by the computer relate not only to this tested system, but also to those excitations. On the basis of theoretical analysis, a mathematical model of CAT's test process has been proposed. It is a sort of Bilateral Poisson Random Point Process, and has two main statistical properties: 1) time interval between two nearest points, and 2) count (score accumulation). A physical model of CAT in block diagram was also given and it has been implemented by hardware and software.

MODÉLISATION D'UN TEST ASSISTÉ PAR ORDINATEUR

Il y a beaucoup de systèmes divers qui doivent être testés fréquemment dans notre travail. Il peut s'agir d'un système social, d'un système machine, d'un système d'enseignement ou d'un système biologique. Dans cette communication, certains problèmes relatifs au modèle de TAO pour le système d'enseignement seront analysés.

Conformément à des objectifs spéciaux, l'ordinateur peut donner certains signaux d'entrée (qui peuvent être appelés "excitations") au système testé, et celui-ci retourne ses signaux de sortie (qui peuvent être appelés "réponses") à l'ordinateur. Pour que celui-ci analyse les informations provenant de ces réponses, nous devons décider des propriétés et des fonctions relatives à ce système testé. En vertu de la loi de causalité, les réponses suivent toujours dans le temps les excitations qui les ont causées. Nous savons que ce retard peut représenter la vitesse de réponse du système testé; en outre, si les réponses correspondent aux attentes préétablies dans l'ordinateur, le système testé est correct; sinon, il est erroné. Les résultats du test et de l'analyse fournis par l'ordinateur ne se rapportent pas seulement au système testé, mais également aux excitations. En se fondant sur l'analyse théorique, on a proposé un modèle mathématique du processus de TAO, une sorte de distribution de Poisson bilatérale de points aléatoires; ce processus a deux propriétés statistiques principales: la première est l'intervalle de temps entre deux points très rapprochés, la deuxième est le compte (accumulation de points). Un modèle physique de TAO sous forme de schéma fonctionnel a également été fourni et il a été mis en oeuvre avec un matériel et un logiciel appropriés.

Model Analysis of Computer-Assisted Test

Wan Jia-Kuo Wu Min-Jin

Institute of Modern Educational Technology
East China Normal University
Shanghai, 200062, China

I) Introduction

The qualified test of colony and individual such as selection of talent, examination of staff and workers, organize in to class for students, entrance examination and athletic competition etc., is a very important work, and we must to do it often and often. But to finish this work with traditional method, it usually takes us a lot of time, nowadays, computer assisted test (CAT) is a modern method of test, it bring us high efficiency and great convenience. In this paper, according to "Item Response Theory" and "Dual-direction Poission Point Process", the mathematical and physical model analysis of CAT are given.

II) A background briefing of CAI model

Response test of drivers for red, yellow and green colour: There are three lamps, red, yellow and green are set up on CAT system, when the red lamp is lighted, the drivers who are tested, must push one of three keys on keyboard as soon as possible, if the key marked with red colour is pushed, we score +1, otherwise, score -1, and at the same time, record delay time of this response(from lighting to pushing). Repeat this test many times, we can get a recording sample, for each driver.

Athlatic competition: We conder that two teams of football are going to compete, at a time, team A get a goal, we score +1, then, team B get a goal, we score -1, so we can make a similar record sample.

Examination: A set of choice type problems can be designed and show one by one to all students in classroom, then the students answer the problems with keyboard as soon as possible. If the answer is corret, we score +1, if wrong, score -1, at the same time, the delay time of responses are also scored. So we still have a similar record sample for each student or each problem.

According to dichotomous score test, we have a lot of examples as mentioned above.

III) Mathematical model analysis of CAT

From three examples as mentioned above, their record samples can be described in figure 1a, where t_{i1} is the response delay time of first problem (football), $t_{i2}-t_{i1}$ is the response delay time of second problem, $t_{i3}-t_{i2}$ is third, and so on. The record samples can also be described by formula:

$$Y_i(t) = \begin{cases} -1 & t=t^{+}_{ij} \\ -1 & t\ t^{-}_{ij} \\ 0 & \text{other} \end{cases}$$

where t^{+}_{ij} and t^{-}_{ij} are the response time of correct answer and wrong answer respectively, both of them are called random point t_{ij} . $i=1,2,\cdots N$, N is the sum totall of students (or drivers), $j=1,2,\cdots K$, K is the sum totall of problems. $Y_i(t)$ are called "Samples of Dual-direction Poission Point Process".

From estimation theory, we have estimation formula of student's (or driver's) feature parameters as follows:

$$\lambda_i = K^{+}_i / T_i$$

$$\mu_i = K^{-}_i / T_i$$

where K^{+}_i and K^{-}_i are the number of problems of ith student's correct and wrong answer respectively, T_i is the sum of ith student's response delay time, so we have:

$$K^{+}_i - K^{-}_i = K$$

$$T_i = \sum_{j=1}^{K} (t_{ij-1} - t_{ij})$$

Therefore, λ_i and μ_i are the average number of correct answers and wrong answers in unit time respectively. $\lambda_i + \mu_i$ is the average speed of ith student's answer, $\lambda_i/\mu_i = K^{+}_i / K^{-}_i$ is called "Ratio of up jump to down jump".

Then we can define the ratio of dual-direction jump rate L_i as follow:

$$L_i = \frac{K^{+}_i / T^{+}_i}{K^{-}_i / T^{-}_i} \cdots \frac{K^{+}_i}{K^{-}_i} \frac{1}{T^{+}_i}$$

where T^{+}_i (or T^{-}_i) is the sum of response

delay time of correct (or wrong) answers, so we have:

$$T_i^+ + T_i^- = T_i$$

$$I_i^+ = \sum_{j-1}^{K_i^+} (t_{i(j-1)} - t_{ij})$$

$$I_i^- = \sum_{j-1}^{K_i^-} (t_{i(j-1)} - t_{ij})$$

The meaning of L_i is different from that of λ_i/μ_i

IV) Physical model analysis of CAT

The process indicated in figure 1, is called "Dual direction equi-amplitude random pulse sequence" in physics, we hope that the positive pulses are more than the negative pulses and the pulse rate is more high, it means that the ith student has better score and response. A block architecture of CAT system given in figure 2 , which can generate the process as mentioned above, was proposed and implemented.

We have made many times of test with this CAT system and analysed those data collected by it, with feature parameters λ , μ, and L, the results are satisfying.

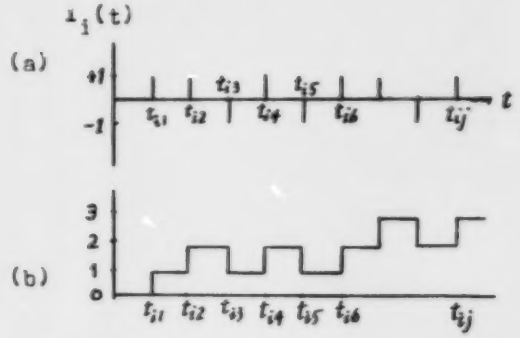


Fig.1, $x_i(t)$ and its accumulation

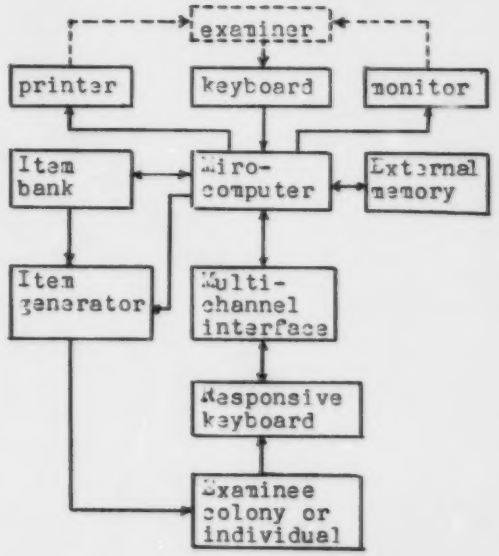


Fig.2, Block diagram of CAT system

ESTABLISHING A CENTRALIZED UNIVERSITY FACILITY FOR THE PROMOTION OF INSTRUCTIONAL TECHNOLOGY

J.M. Fritz, G.A. Jordan, J. Webster
School of Computer Science
University of New Brunswick
Fredericton, NB

The purpose of this paper is to report on the experience gained during the establishment of a Centre for the Promotion of Instructional Technology at the University of New Brunswick. The authors were given a mandate to set up a resource facility, accessible to faculty and students from all disciplines on campus, to develop and promote the use of instructional technology in teaching, and in particular to foster ongoing collaborative work in developing technology-based teaching aids for a wide variety of areas. The installation of a centrally located resource centre, equipped with Macintosh microcomputers, was viewed as providing the environment necessary to: 1) develop computer aids for both classroom use and self-instruction or tutorial use in an open lab setting, 2) demonstrate the use of computer aids to all professors who are interested or curious about the possibilities, and 3) provide technical assistance for professors to develop their own computer aids.

One of the most illuminating results of our initial overtures to the university community, when soliciting input for workshop needs, was the identification of faculty members from a diversity of disciplines who already had some very creative ideas for computer aids to teaching, but who had not pursued their ideas for perceived lack of university support and resources. We were able to act as a communications link to bring together people and resources and to facilitate equitable and optimum use of resources.

ÉTABLISSEMENT D'UN CENTRE UNIVERSITAIRE CENTRALISÉ POUR PROMOUVOIR L'ENSEIGNEMENT DE LA TECHNOLOGIE

Le but de cet exposé est de faire un compte rendu de l'expérience acquise au cours de l'établissement de centre de promotion de l'enseignement de la technologie à l'Université du Nouveau Brunswick. Les auteurs de ce projet avaient pour mandat d'établir un centre de documentation accessible aux membres de faculté et aux étudiants de toutes les desciplines du campus en vue d'accroître et de promouvoir l'emploi de méthodes technologiques dans l'enseignement et particulièrement, de favoriser le travail d'équipe continu pour developper des méthodes d'enseignement a base technologique visant de nombreaux domaines. Nous avons ainsi envisagé l'installation d'un centre de documentation central et équipé de micro-ordinateurs Macintosh dans le but de procurer un cadre essentiel pour pouvoir: 1) developper les aides-ordinateurs facilitant en même temps l'enseignement individuel et de groupe ou à l'usage des moniteurs dans les laboratoires publiques, 2) expliquer le fonctionnement des aides-ordinateurs à tous les professeurs y manifestant de l'intéret ou de la curiosité, et 3) fournir de l'aide technique aux professeurs désireaux de développer leurs propres aides-ordinateurs.

Lors de nos premières tentatives faites en vue de soliciter des suggestions pour la création d'ateliers, un des résultats les plus marquants a été d'identifier des membres de faculté appartenant, a différentes disciplines qui avaient fait preuve auparavant d'idées créatrices en vue de démontrer l'importance de l'ordinateur dans l'enseignement mais qui n'ont pas poursuivi leurs idées à cause du manque de documentation ou d'encouragement de la part de la communauté universitaire. Nous avons été en mesure de servir de lien de communication afin de réunir les personnes et la documentation et de facilites un emploi équitable et optimum de ces ressources.

ESTABLISHING A CENTRALIZED UNIVERSITY FACILITY FOR THE PROMOTION OF INSTRUCTIONAL TECHNOLOGY

Jane M. Fritz, Glen A. Jordan, John Webster

University of New Brunswick
Fredericton, New Brunswick, E3B 5A3

The University of New Brunswick formally endorsed the use of microcomputers as aids for classroom teaching with its financial support to establish a central facility for the promotion of instructional technology. It is hoped that our experiences to date with the fledgling Centre for the Promotion of Instructional Technology may prove useful to others whose institutions have a similar history of isolated instances of efforts to obtain or use computer-based support material.

Background

The history of instructional technology at UNB could be best described as fragmentary. Those instructors who had personal knowledge of computing and access to a computer were in a position to be able to appreciate the potential for computer-based teaching aids; a few enthusiastic people in a diversity of disciplines developed and used such aids in isolation. Others heard of existing CAL material appropriate to their curriculum at conferences, in journals, or from colleagues at other institutions, and wondered what they were missing. The overriding theme was that, because there was no visible driving force on campus, possibilities for cooperation and collaboration for acquiring hardware and software for instructional purposes simply did not present themselves.

Strategy for the introduction of microcomputer labs has been, by and large, one of crisis response. In a time of tight resources and great demand for microcomputers, the allocation of new resources went first to labs for programming courses, and then for business administration courses and for education students for selected course work. With each addition of new resources, the intent has been to serve as an integral part of course curriculum, as opposed to use for computer-aided instruction or any innovative use of the computer to reinforce a non-computer related concept in any curriculum. As computer resources proliferated, this scenario continued, with new resources being installed to accommodate such curriculum requirements as computer-aided drafting, course projects, etc. All of the labs to date have networked micros; programming and business administration labs consist of IBM PCs, while the Education Faculty has an Apple IIe lab for their "computers in the classroom" courses and an IBM PC lab for their business education courses. All but the education labs are open rather than restricted to scheduled use.

An exception to this restricted use of computers to areas in which they are an intrinsic part of the curriculum was the establishment of a classroom of microcomputer stations in the Department of Forest Resources. This classroom, or dedicated lab, is used to reinforce the philosophy of integrating the computer across the curriculum. To that purpose this resource is used in many Forest Resources courses for such applications as report-writing, statistical analysis and modelling, and computer-assisted learning (CAL). This lab serves ten undergraduate courses that involve more than 22,000 student-hours of formal CAL activities and use 17% of available laboratory time. As well, a full-time CAL specialist, dedicated solely to courseware acquisition and development, is available to faculty, along with an extensive courseware library. Prof. Jordan, who was instrumental in the realization of computer-based learning and teaching within the Department of Forest Resources, became convinced of the potential for computer-based teaching aids for a wide variety of disciplines. Accordingly, he spearheaded a small group of enthusiastic faculty and staff members to establish a central resource facility that would foster the use of instructional technology across the campus.

The Facility (CPIT)

The Centre for the Promotion of Instructional Technology (CPIT) was established with the support of the university administration to provide a central site to:

i) allow instructors to be exposed to the potential of instructional technology,

ii) allow improved access to equipment, and

iii) foster collaboration in the development of in-house materials and courseware.

The Centre, housed in a centrally located lab in the Department of Audio Visual Services, has been equipped initially with six Macintosh microcomputers including large-capacity hard disk drives, a laser printer, a digital scanner for graphics, a videodisk player, and a projector attachment for overhead computer display. Faculty and staff are encouraged to make use of this facility in one or both of the following ways:

i) Through workshops offered by Centre organizers. These workshops are directed particularly at faculty and support staff from one academic department at a time.

ii) Through open access for experimentation and practical use. The lab is always open during regular university office hours. Expert help is available on a scheduled basis for such activities as: preparing overhead material, picture scanning, HyperCard applications, and straightforward word processing for classroom handouts.

Initial Projects: Workshops

The first activity introduced under the jurisdiction of the Centre was an offering of one-day workshops aimed at bringing together both teaching and secretarial staff to explore the numerous possibilities afforded by microcomputers and laser printers in managing graphical and textual information for course use. The response to these workshops has been very positive. Academic units that have participated to date include: nursing, sociology, library, biology, chemistry, and physical education.

Workshops aimed at the development of educational software for individual lectures are the next item on the Centre's agenda, particularly the use of HyperCard as the delivery metaphor. A demonstration of the use of software in a classroom setting was offered recently to whet the appetite. An invitation was sent to all faculty members to attend a "live" lecture where a microcomputer interfaced to an overhead projector was employed to enhance a conceptually difficult lecture. It was emphasized that computer-based techniques, in this case the navigation through a HyperCard stack, need not be specific to any one discipline, but in fact offer options for delivery and demonstration that are not possible using a chalkboard and/or conventional overhead projectors.

Initial Projects: Software Development

A long-term project being carried out jointly by the founders of the Centre (Glen Jordan, Forest Resources; Jane Fritz, Computer Science; John Webster, Audio Visual Services), along with participants from Surveying Engineering, involves the research, development, evaluation, and demonstration of an innovative educational software approach in teaching and learning. The approach involves hypermedia software (and appropriate hardware) characterized by two-fold utility:

i) as an in-class teaching aid, replacing for the most part traditional blackboards, overheads and other audio-visual devices, and

ii) as a student-driven learning aid, not unlike library reference material, available upon demand outside the classroom.

The two parallel areas of concentration at the moment are the development of educational software modules for spatial analysis and geographic information systems (GIS), and the development of low level tools to assist faculty (who may have no computer skills or experience) to utilize the graphics and sounds of hypermedia and interactive video in teaching their various subjects. This work is being supported with developmental equipment from the Apple Canada Education Foundation.

The Centre will play an important part in the evaluating phase of this project. All instructional packages developed under the mandate of this project will be evaluated for their effectiveness through the use of a research laboratory; in the preliminary phases, the Centre will act as the research laboratory.

Although the software modules under development are directed at specific courses, project participants are using the experience gained from developing the context-specific material to build a simple framework that would allow unexperienced instructors in any discipline to quickly and easily generate interactive visual diplays for classroom use. To be truly more useful than traditional visual devices, such displays will have to be not only interactive but 'smart', responding to unstructured student input as well as foreseen manipulation by the instructor.

Observations and Conclusions

The establishment of the CPIT has not solved many problems. In fact, it has created more problems than it has solved, in that we have stimulated the desire for even more resources. Once it was advertised that such a facility was operational, requests to serve wide ranging needs began to trickle in.

An instructor in geology is interested in using a videodisk player, controlled by a microcomputer, for classroom presentations. Of course, both still and moving pictures are available, and the microcomputer allows one to organize and reorganize any presentation with ease, mixing still and moving pictures from the same disk. Once this potential is demonstrated, it is anticipated that demand for the acquisition of disks will increase dramatically.

An instructor in classics requested the purchase of courseware for a Latin course, which is now being used and evaluated. Someone in biology wants to explore all available material in their field and possibly develop some customized lessons. An English professor has developed courseware for a PC and never knew who to push for satisfactory computer access for Arts students so that his software could be used. The Centre has become the advocate, or ombudsman, for these enthusiastic but seemingly alienated teachers.

It hasn't taken long for the longstanding problem of equipment incompatibility to rear its ugly head. CPIT was equipped with Macintosh computers and peripherals for several reasons; ease of use for non-computer oriented users, availability of HyperCard and HyperTalk, Macintosh expertise on the part of the Center founders, and ease of development of satisfying graphical displays. However, PC microcomputers are widespread on campus, particularly in labs, and so requests for PC supported materials and peripherals such as compatible overhead projector attachments are becoming common. CPIT views this as a positive situation in that we are being identified as a facilitator and thus the current needs and uses across the campus are starting to be identified at the same time as we are generating new needs and uses.

Even in this early stage of development we believe that the exposure the Centre has been able to give to the possibilities of instructional technology has been significant. An identified driving force for the promotion of computer-based learning resources ensures improved communication between innovators and users across the campus. Slowly, the full complement of Macintosh and PC based resources will be acquired, hopefully in a more coordinated and rational manner than would be possible without the focus the CPIT can offer.

A MODEL FOR THE APPLICATION OF INSTRUCTIONAL TECHNOLOGY IN A LITERACY AND EMPLOYMENT PREPARATION PROGRAM

C. LeBlanc
Industrial Cape Breton
Youth Employment Opportunity
Sydney, NS

D. Fugler
Control Data Canada Ltd.
Toronto, ON

The Industrial Cape Breton Youth Employment Opportunity began in 1984 to provide employment preparation training for severely employment-disadvantaged youth. At that time the unemployment rate in the area was one of the highest in Canada. Youth were particularly hard hit due to their low education, limited skills and defeatest attitude. Given this, it was almost impossible for youth to find work, particularly in an economically depressed area.

The innovative program incorporates computer-assisted and managed instruction in conjunction with group discussions and individual counselling. The computer-based components include: Basic and High School Skills, Life and Work Skills and Job Search. The use of CAL enables youth who have not been successful in traditional learning environments to develop literacy, social and job search skills. More importantly, youth acquire the confidence to choose, obtain and retain employment. To-date, 86% of participants who had not been able to obtain employment prior to this training are now working. The success of the program is based on the combination of CAL, with group activities and the support of a dedicated, competent staff.

The paper presents the design of the program components, the evaluation methodology in terms of academic achievement, job readiness, and attitudes toward self, school, computers and employment.

UN MODÈLE D'APPLICATION DE LA TECHNOLOGIE PÉDAGOGIQUE DANS UN PROGRAMME DE CONNAISSANCE DE L'ÉCRIT ET DE PRÉPARATION À L'EMPLOI

L'Industrial Cape Breton Youth Employment Opportunity a instauré en 1984 un programme de formation pour préparer au travail les jeunes qui sont grandement désavantagés sur ce plan. À cette époque, le taux de chômage régional était l'un des plus élevés au Canada. Les jeunes étaient plus durement touchés en raison de leur faible niveau d'instruction, de leurs compétences limitées et de leur attitude défaitiste. Il était par conséquent presque impossible pour les jeunes de trouver du travail, particulièrement dans cette région économiquement faible.

Le programme innovateur instauré offre un enseignement assisté et géré par ordinateur avec des discussions en groupe et des consultations d'orientation individuelles. Le logiciel porte sur les connaissances élémentaires, les compétences en matière de travail et la recherche d'un emploi. L'utilisation de l'EAO permet aux jeunes qui ont échoué dans les milieux d'enseignement traditionnels de s'alphabétiser, de s'adapter à la société et d'apprendre à chercher un emploi, et surtout, leur donne confiance pour choisir, décrocher et conserver un emploi. Jusqu'ici, 86 % des participants qui n'avaient pu obtenir un emploi avant cette formation travaillent maintenant. Le succès du programme est dû à la combinaison de l'EAO avec des discussions de groupe et à l'appui d'un personnel dévoué et compétent.

Cette communication présente les composantes du programme, la méthodologie d'évaluation, laquelle est basée sur le rendement scolaire et l'aptitude à l'emploi, et l'attitude du participant envers lui-même, l'école, l'ordinateur et l'emploi.

A MODEL FOR THE APPLICATION OF INSTRUCTIONAL TECHNOLOGY IN A LITERACY AND EMPLOYMENT PREPARATION PROGRAM

Coleen LeBlanc & Dorothy Fugler

Industrial Cape Breton Youth Employment Opportunity
Control Data Canada, Ltd.

SUMMARY

Industrial Cape Breton Youth Employment Opportunity (YEO) is a computer-based employment preparation program sponsored by the Industrial Cape Breton Board of Trade and funded through the Department of Employment and Immigration.

In 1984, YEO was established to meet the needs of severely employment disadvantaged youth. At that time, the Industrial Cape Breton area was experiencing one of the highest unemployment rates in Canada. In such an environment, it was almost impossible for young people to find and retain employment.

An analysis of the target group identified a number of barriers to gaining employment. These included: low educational levels, low self esteem, lack of or limited work experience, inappropriate communication skills and underdeveloped social skills. In addition, many youth expressed a feeling of despair knowing that they lacked the social, academic and job specific skills necessary to gain employment. Prior to the introduction of this program, there were virtually no options for youth with these barriers but to continue the cycle of unemployment or to leave Cape Breton.

The training needs of the target group included: literacy, life/work skills, and job search. Consideration was given to various delivery methods. A review of programs designed for disadvantaged youth indicated that computer-assisted learning appeared to be particularly well suited.

The Control Data PLATO computer-based learning system was selected to provide the testing, instruction and management of most of the academic upgrading and the initial lessons in life/work skills and job search. In addition to the computer-based programs, a significant amount of time was scheduled for individual counselling and group support.

The results of the program have been impressive. In summary:

93% of participants have completed the program

87% of those who completed are now employed

89% of those employed are still in the same or better job.

These results become even more significant when you consider that over the past four years, one-third of the participants have never been employed and those who had jobs had experienced over two years of unemployment prior to entering the program.

PROGRAM DESCRIPTION

The goal of YEO is to provide pre-employment training for severely employment disadvantaged youth in the Industrial Cape Breton area and place them in employment or further employment-directed training.

An analysis of participants over the last four years indicates the following characteristics:

Average age	20
Average grade level	10
Average years out of school	3
Percent never employed	34%
Average years unemployed	2.4

In addition to the academic and employment barriers, many participants must also cope with one or more of the following social barriers: criminal history, social assistance recipient, history of drug/alcohol abuse, single parent, visible minority.

The YEO approach to employment preparation is individualized and experiential. Following assessment, an individual

Employment Preparation Plan is developed. Each participant starts the academic upgrading component at the appropriate level.

The PLATO Learning System provides computer-based testing, instruction and management. The instructional programs are competency based. These competencies are defined in terms of instructional or learning objectives. Each objective must be mastered according to specified criteria before the learner can proceed to the next one. Diagnostic testing is delivered through the computer. The computer-managed instructional system relates the test results to the instructional modules enabling participants to move directly from testing to immediate feedback to instruction. The instructional materials are delivered primarily through the PLATO Learning System, supplemented by text, audio-visual material, and tutoring as required.

Participants progress through the instructional material at their own pace, receiving assistance in selecting the path of instruction that is most appropriate to their needs. Records on each participant are maintained by the computer. These assist staff in tracking progress and identifying areas for individual tutoring. In addition to the daily support, counsellors meet weekly with each participant to review/revise the individual Employment Preparation Plans.

During the initial phase of training, participants spend approximately one half of their time in individualized academic upgrading which includes Basic Skills Mathematics, Reading, and Language Arts and High School Skills Mathematics, Reading, Writing and Science. The other half is spent in life/work skills which include a combination of computer-based, individual and group learning activities. With the knowledge, skills and confidence gained through the individualized instruction and counsellor support, participants are able to engage in group discussions and role plays to further develop employability skills.

Once the participants have achieved their academic and life/work skills goals, they progress to the final phase of employment preparation.

Job search training is delivered through the Control Data Job Search Program. The program provides instruction and practice in the techniques of self-directed job search which assist participants in obtaining employment. The Job Search Program includes instruction in all aspects of conducting a successful self-directed job search:

- practice in contacting employers and being interviewed through role play and simulation exercises;
- motivational support through individual and group counselling and one to one peer activities; and
- systematic job search through a specific daily routine of prospecting for jobs, contacting employers, scheduling and attending job interviews.

As with the life/work skills program, training is delivered through a combination of computer-based lessons and small group sessions. Individual counselling and group support are provided to guide participants through the program and assist them in developing positive attitudes toward work and increased self confidence in their own employability.

In addition to the structured learning activities, employment preparation is achieved by participants through the experiential nature of the overall program. Rather than being students, individuals are treated as employees. The training provided in this job is designed to enable them to obtain another job. There is a subtle but significant distinction between being paid as an employee versus receiving a training allowance. Indeed, for many participants, employment at YED is their first job.

EVALUATION

All components of YED are measured in order to evaluate participants' progress and to determine the effectiveness and efficiency of the program. Formative and summative evaluation are used as consideration is given to both the process and the product. Specific data are collected from participants upon entering the program, during each training component and upon completion. Participants' progress is determined by analyzing these data. Overall program effectiveness is also measured in this way.

Assessment of the academic level at which participants are functioning is determined through the use of the Canadian Adult Achievement Test (CAAT). At the end of the academic upgrading and life/work skills components, a post-battery of CAAT is administered. Academic achievement is measured by the difference in functional grade levels.

It was apparent through both observation and feedback that a major change for participants is in their attitudes toward themselves, learning, and their ability to find and keep a job. An attitude questionnaire designed to address these areas is administered early in the program and repeated at the completion of the academic and life/work skills components.

It is interesting to note that a significant shift to a more positive attitude has been reflected in all areas measured.

Participant evaluation is ongoing and includes: testing and progress checks for all computer-assisted instruction, pre/post vocational inventory testing, weekly progress reviews with Employment Counsellors, and feedback through the various individual and group activities.

On completing the program, participants fill out a Program Evaluation form. This allows participants the opportunity to comment on all aspects of the program.

Of course, the ultimate measure of effectiveness of the program is in the placement and retention rate of program participants. For a period of one year after placement, an ongoing dialogue continues between YED staff, participants and their new employers. Post program follow-up visits have proven to be an effective means of evaluating the transfer of newly acquired skills to the job.

Yearly follow-up is conducted with all former participants to measure the long term effect of the program.

RESULTS

The following summarizes results to date in the key performance indicators:

Participants (15 current)	Number	Percent
Started	244	-
Completed	224	93.7
Employed	194	86.6
Retained	173	89.2
Average Attendance -		97.8

Academic Upgrading	Pre	Post	Change
Mathematics	9.8	10.6	1.6
Reading	10.8	11.6	.8
Language	10.6	11.7	1.1
Average	10.0	11.2	1.2

It is important to note that while the average pre-test grade level is 10, the actual range has been 3.5 to 12. It is generally agreed that functional literacy in today's society requires a minimum ninth grade level while those functioning below a fifth grade level are considered illiterate. A review of the pre-entry functional levels in numeracy and literacy indicated 47% functionally illiterate in reading and/or language arts; 31% functionally illiterate in both reading and language arts; and, 78% functionally illiterate in mathematics.

The preliminary program design was based on an expectation of the need for some remediation prior to the intensive employment preparation. In fact, it became apparent that literacy would have to be a major component of the program. This emphasis on literacy led to the encouragement of the use of General Education Development (GED) exams as a confirmation of academic achievement. To date, 59 participants have written the GED examinations. Of these, 48 obtained Grade 12 in the 5 subjects, the others achieved a mixture of grade levels. In each case, there was improvement over the last grade attained at school.

While the statistics are impressive, the impact of the program has been felt in other ways as well:

Severely employment disadvantaged youth have developed the skills and confidence necessary to choose, obtain, and retain employment.

There has been a significant rippling effect in terms of economic development. Youth from the program are now employed and contributing to the economy.

The attitudinal changes experienced by the participants have spread through the community. There are jobs in Cape Breton. It is possible to escape the cycle of social assistance. Cape Breton's youth do not have to leave home to become productive members of society.

The program has raised local expectations of the benefits to be gained from innovative upgrading programs delivered outside of an institutional environment.

In summary, YED is exemplary by the standard measurement criteria of employment preparation programs: attitudinal and behavioral change, improvement in grade level and usable knowledge, percent finding and retaining work, and cost per graduate.

IMPLICATIONS

The Industrial Cape Breton Youth Employment Opportunity has achieved a remarkable level of success. The focus of this study has been to determine the factors influencing the effectiveness and efficiency of the program.

A review of the data and discussions with staff and participants indicate that there are four critical success factors: individualization, computer-assisted learning, comprehensive training and competent, caring staff.

The focus of YEO is on meeting individual needs. From initial assessment to the individual Employment Preparation Plan to tailored instruction, all components are adapted to the needs of each participant.

The capability of the program to tailor instruction is based on the use of instructional technology. The comprehensive computer-based learning system provides individualized, self-paced instruction which adapts to entry level. Diagnostic testing with immediate scoring and assignment of appropriate learning activities enables participants to move rapidly through the required lessons with minimal instructor intervention. The counsellors' time can be spent in individual tutoring as required. The management system reduces the administrative time normally consumed in record keeping. Again, the counsellors can focus their efforts on providing individual support.

As indicated previously, many of the participants have not been successful in more traditional learning environments. The use of computer-assisted learning has made a significant difference in the participants' attitudes toward learning and to themselves as learners. Privacy, absence of peer pressure, immediate feedback and the opportunity for a successful learning experience are the benefits most often cited by participants.

Perhaps most significantly, for the first time participants become responsible for their own learning. This may be the most critical skill they learn since technological advances reinforce the need for life-long learning.

The third success factor relates to the comprehensiveness of the training components. The program provides training in all of the skill areas required for employability. Participants become confident that successful completion of their Employment Preparation Plan can and will lead to employment.

It has become clear that severely employment disadvantaged youth require extensive support in order to develop the positive self image required to complete an intensive training program. The staff at YEO are committed to the success of each participant. Comments regarding the knowledge and support of the staff consistently appear on program evaluations.

It is appropriate to conclude with a brief description of the current direction of YEO. The technology-based model has been influenced by and become a catalyst for technological change. The use of computers for instruction has led to a certain degree of computer literacy. During the initial years of the program, the use of microcomputers grew in the community. Participants responded well to computer-assisted learning and as a result became interested in learning more about computers. Given the community's acceptance of YEO, the increased use of microcomputers in business and industry and the interest of participants, it seemed appropriate to build on the technological basis of the initial program.

Based on a survey of local employers to determine training requirements, an Office Automation Training program was designed to be incorporated with the current employment preparation program. The training content includes office procedures, clerical skills and specific microcomputer applications. Training includes computer-based, individual and group learning activities.

After a successful pilot, the expanded program was introduced in 1987. Results to date include 90% placement with 100% retention.

'MENU': APPLICATION TOOLS FOR AN EDUCATIONAL LANGUAGE ENVIRONMENT

P. Beam, R. Siemens
Dept. of English
University of Waterloo
Waterloo, ON

To facilitate computer operations for novices, we have created eleven functions which provide a powerful, efficient computing environment. Easily accessed, they organize operations which students must perform in the completion of an on-line course in analytic criticism.

The basis is MENU, a student-access mechanism and screen. From function keys students select the various operations, ESSAY SCRIPT, the assignment creation template, automatically generates footnote and bibliographic entries and the formatting of titles and pages of assignments. HANDIN and MARKER ensure the instructor enjoys correct document formatting, grade recording and class tutorial generation. RESUME enables students to produce a complete and properly formatted resume by entering pertinent information via computer prompts. HELP files for specific documents, concepts and styles of writing, created by students using the system, instruct new users in the subject matter as well as explaining concepts and formats. Other functions include the interactive analysis of literary materials, NEWS, MAIL and communications facilities.

This grouping retains all the options and power of the CMS environment. Its simplicity acts as a bridge to permit computer novices to use and prepare sophisticated assignments and to communicate extensively, even as they learn basic computer processes.

'MENU': OUTILS PRATIQUES DANS UN ENVIRONNEMENT D'APPRENTISSAGE

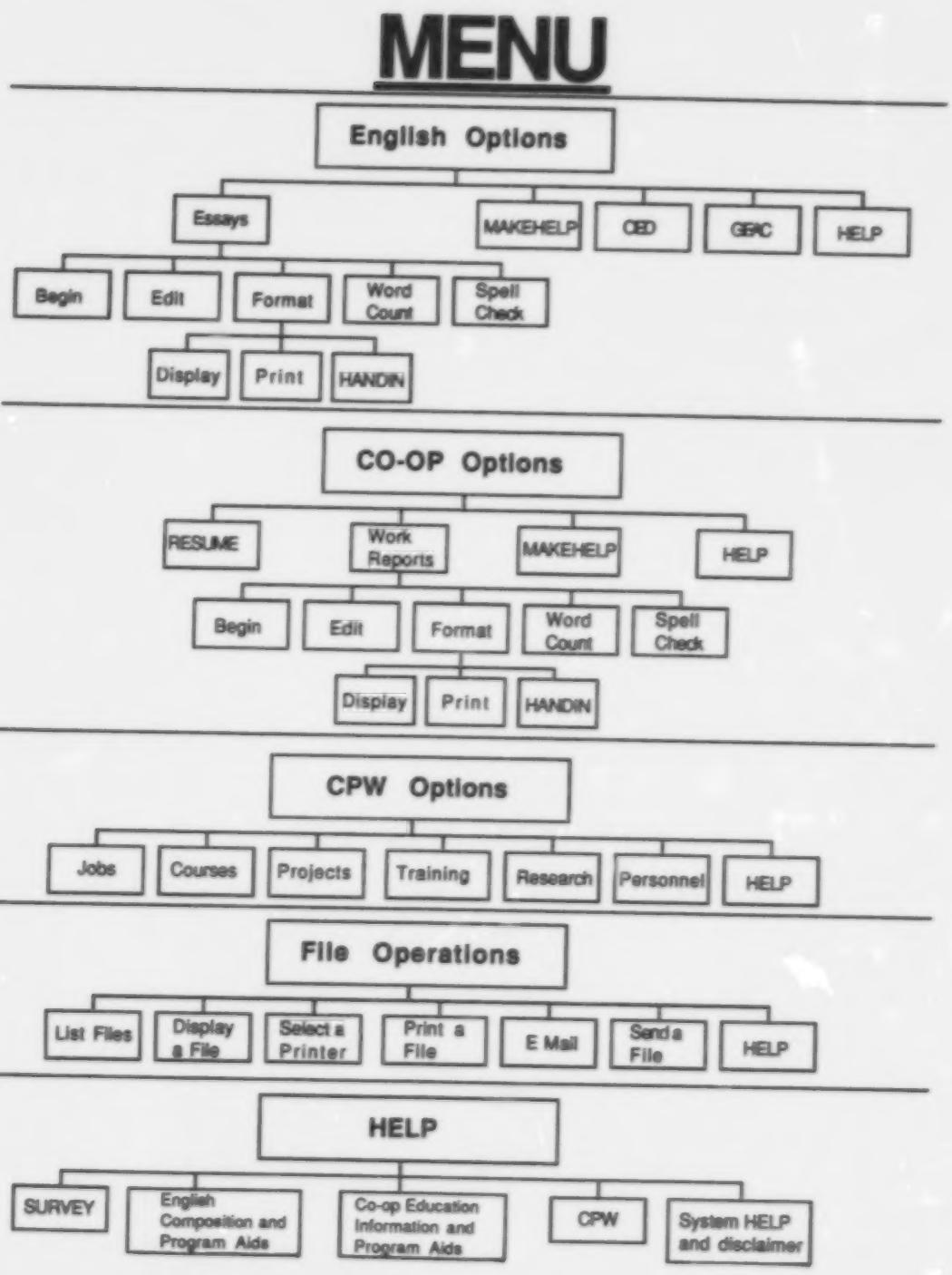
Pour que les débutants exécutent plus facilement les opérations informatiques, nous avons développé un groupe de onze fonctions qui constitue un environnement informatique puissant et efficace. Faciles à appeler, elles organisent les opérations demandées aux étudiants qui suivent un cours en direct sur la critique analytique.

La fonction de base est MENU, à la fois un écran et un mécanisme conduit par les étudiants. Ces derniers en pressant sur les touches de fonction choisissent diverses opérations. ESSAY SCRIPT, le modèle à suivre pour la création d'un exercice, génère automatiquement les renvois en bas de page et les entrées bibliographiques et met en forme les titres et les pages des exercices. HANDIN et MARKER font en sorte que l'instructeur est assuré du bon formatage des documents, de l'enregistrement des notes et de la génération de l'enseignement à la classe. RESUME est une autre fonction qui permet aux étudiants de rédiger un résumé complet et ayant la forme exigée, en donnant seulement les bonnes réponses aux questions de l'ordinateur. HELP met en fichiers pour certains documents les notions et les styles de rédaction créés par les étudiants à l'aide du système; de plus cette fonction donne aux nouveaux utilisateurs les instructions sur la matière et explique les notions et les formats. En plus, on trouve dans ce groupe des fonctions d'analyse interactive des documents bibliographiques, NEWS, MAIL et des programmes de communication.

Ce groupe de fonctions conserve toutes les options et toute la puissance de l'environnement CMS. La simplicité de ces outils permet aux débutants de réussir à faire des exercices compliqués grâce à un dialogue constant avec le système, en même temps qu'ils apprennent les rudiments des processus informatiques.

'MENU': Application Tools for an Educational Language Environment

Paul Beam & Ray Siemens,
Department of English, University of Waterloo.



INTRODUCTION

To facilitate computer operations for second-year Honours Co-op English students while they simultaneously learn to work on a mainframe system and use it for information and assignments in their courses, we have created a group of programs which provide a powerful, efficient work environment. They are easily accessed and organize operations which students must perform in the completion of their on-line English courses.

ORGANIZATION

These functions are grouped into five categories: English writing functions; communication and information for the Centre for Professional Writing; Co-op Education procedures and communication; System Operations; and On-line Documentation and Services. Devices helpful for the research, writing, and formatting of a student assignment are incorporated by the English Functions. The CPW level offers information on projects and contracts in the Professional Writing program. Co-op Information provides a forum for the Co-operative education program within which students can investigate potential jobs, contact program co-ordinators (and some employers), develop and submit resumes and work reports from templates and models, and talk with each other about work experiences. Students can perform a number of basic operations, such as MAIL and MESSAGING, in a simplified format within the System Operations menu. Each stage of the process is augmented by a package of concise, PF-key accessed documentation, designed for the naive user, describing both the function of the system and the options available at each level.

The programs existing within these categories simplify course and program administration, assist in computer communication and language proficiency, and offer background and teaching aids for subject materials. The package provides the structure for a congenial on-line course to be conducted and provides efficient, easy to use, communications and writing devices at all stages throughout the Honours English program.

The user accesses all these devices through MENU, which organizes the tools, options, and points of address for a naive user. The MENU interfacing device couples specific language-related functions which we have developed, like RESUME, to those already present on the CMS system, like SPELL. We permit experienced students to by-pass 'cradling' structures when their familiarity with CMS makes more direct commands more efficient.

By offering an alternative to the traditional command-line oriented CMS environment and providing concise on-line documentation via a PF key, the system assists naive users to perform tasks successfully during their initial experiences with the system. These simplifications of processes are accurate, but limiting. "MAIL, for instance, in its naive format, presents a list of class members, instructors, and co-op advisors among whom the user may browse and select via the TAB and <CR> keys. Instructions give not only the steps but also their options and consequences. At any point, the user may exit and use the system's version of "MAIL", which is more direct, but also more cryptic. First-time users, however, are assured an accurate and successful initial application, from which they may venture to further domains and more distant contacts.

ENGLISH FUNCTIONS

Within the English Functions group lie several different sets of tools aimed at assisting in the process of assignment research, writing, and submission.

In addition to conventional word-processing capabilities, students may take advantage of several Computer Aided Learning (CAL) tutorials such as ROLOCK, a fully developed CAL tutorial on Alexander Pope's RAPE OF THE LOCK. Class tutorials are generated by the instructor's MARKER tool and can also be placed on-line, as can the instructors' lecture and class notes, for student use at any point in the course. There are many literary texts and passages in an on-line format as part of a student's resources for models, analyses and testing.

At the time of writing, we are linking this structure to the University of Waterloo's GEAC library database to allow students to easily examine the contents of the University

library in the immediate context of their assignment. As well, students have access to both the New Oxford English Dictionary and the Advanced Learners' Dictionary for reference and for concordance checking of terms and in both texts and assignments.

Once background research is done, the student uses ESSAY SCRIPT and HANDIN to assist in writing and submitting assignments. ESSAY SCRIPT is a template of the SCRIPT formatting language. Because the formatting details are assumed by the template, students can spend more time concentrating on the content of their work and instructors can count on receiving properly formatted student assignments for marking. Even the style of footnotes and bibliography is simplified by ESSAY SCRIPT with both 'help' models and PF-key insertion of information into pre-defined format codes to assure correct display on screen and in printed versions. The entire course is conducted 'on-line' with assignments created, submitted, marked and incorporated into tutorials for class distribution in electronic format only, though students may make print-outs of course notes, mail, projects, models and their own writing at any point. At the conclusion of the writing process, students can check their assignments with a word counter, spell checker, and style monitor. They may also view their submissions for final changes on the screen to ensure correct formatting and display before submission.

When the assignment has been edited and is ready for marking, users call HANDIN. This properly formats the assignment and sends it to the instructor's account to be marked. It also appends information about the student's identity and the assignment number to the file, so it can be easily identified for its grade and subsequent return.

Instructors use a program called MARKER to assess assignments on-line. It provides facilities to permit easy movement through the student's text, the saving of segments of student essays for later incorporation into a class TUTORIAL, assignment of recurring comments like SPELLING ERROR or DOCUMENT to be generated by functions keys which the instructor can preset for the kinds of comments most frequently used and the automatic entry and storing of marks and averages in the class list and on individual papers.

Throughout the entire process, at any point in the system, a function key gives the user specific on-line documentation of matters as diverse as rules of the text editor, assistance in composition and the effective operation of essay tools.

In the event that students cannot find proper documentation on an aspect of the system, they have the option of either informing the instructor of the deficiency by electronic MAIL, or, if they have a resolution, they may invoke MAKEHELP to describe the problem they encountered and how they resolved it. The instructor can evaluate the solution and place the new help-screen on the system for the benefit of all users. In this way, documentation is kept current by students themselves and is directed by instructors into appropriate locations to help others.

Announcements can be made to the class, either by a NEWS function which displays messages when students log onto the system, or by electronic mail to individuals or user-defined groups. Whole tutorials can be run and transferred in finished form to the B disk for class use.

CO-OP OPTIONS

For the student involved in the Co-op program, this section of MENU provides a RESUME creation program, a template similar to ESSAY SCRIPT, which properly formats WORK REPORTS and a series of topical information screens for INTERVIEWS, EMPLOYER RESPONSIBILITIES, SALARY, and the like. These are augmented and revised to remain current.

RESUME allows students who are not familiar with the computer's formatting codes to create a complete and correct document. They enter and revise information sequentially as they are prompted by the computer. The document is then tagged with the proper formatting code to be laser-printed and sent directly for the Co-operative Education job interviews that all users face each term.

WORK REPORT writing is also made simpler by using the template provided in this section. This Xedit file contains the proper formatting information and topic areas required to write a full work report in accordance with the

University's guidelines. Additional Co-op information exists in the on-line documentation area of MENU where students can get assistance on aspects of the Co-op program and on the use of computer tools relevant to it.

SYSTEM OPERATIONS

The devices at this level, including the interfacing program MENU, operate within the CMS environment and in effect combine functions already present on the system. By invoking them students can also take advantage of the many programs and tools CMS offers, including complex file operations, inter-university and global electronic mail, file transfer and messaging. Some of the more commonly used operating system functions are accessed in the same way as other MENU functions. These include LISTING files, PRINTING, MAIL, messaging, and file transmission. In other more sophisticated areas, the student must move from this environment to the increasingly complex operations of the CMS system itself but this brings corresponding knowledge and power.

To familiarize new users with the functions and devices contained in MENU, a "walk-through" of the many options offered by MENU is available with a program called SURVEY.

FUTURE DIRECTION

Our intentions are to develop this module and to make it the learning vehicle for approximately two hundred English students in the fall term of 1989. It will be available for the cost of transmission to any academic units wishing to use or experiment with all or parts of it. Its virtues are: as a comprehensive main-frame training device for large numbers of naive users who need to quickly employ computer techniques for their writing, communications, information and administration requirements during the term, even as they learn about the capacities and techniques of a large computer system. Information and aids in their accounts help them in assignments and in understanding of subject matter; inter-active lessons provide advice on, and assessment of, their analyses of text and of their written expressions of those analyses. Communications appear initially to be present for purely social and administrative reasons, but the instructors' applications of on-line tutorials and project groups make this a major source of learning and information exchange in its own right.

From their understanding of system components, students quickly come to use their accounts to evaluate existing programs and databases, to implement and integrate new materials and devices and to learn and write about additional programs for their own, their groups' and the system's use. In this way the structure acts simultaneously as teacher, repository, contact point and development tool for students from naive, initial users to knowing fourth-year students with extensive experience in cybernetics and analogous devices in commerce and industry. In fact, this structure has three important contact points to systems outside the program and university structures: the Center for Professional Writing lists projects in process (on some of which Co-op students work directly from their English accounts in work and school terms); work reports developed using the work-report generator mentioned above are available as information for prospective applicants about the particular job, as well as models of the report-writing process itself. Finally, several companies and institutions share contacts across the mail system directly and provide links to their own databases and projects on which the Center and individuals collaborate.

The project is expansive both in the numbers of users it serves and in the variety and complexity of programs and databases it accesses. It has been designed to include new and increasingly interactive options as we acquire or create them. We plan to extend it, in whole or in part, to other academic and administrative areas of the University. It draws on three resources unique in combinations and results: Waterloo research expertise in programming and main-frame applications, the real experience and skills areas of senior students and instructors and the interaction between students and the employers with whom they work in increasingly sophisticated projects during co-op terms. The device which trains them ultimately becomes the agent of their own research and academic expression, even as their work serves to inform and train succeeding classes of their peers.

CATO - COMPUTER ASSISTED TOPICS

P.R. Watterson
Inservice Division
Jordanhill College of Education
Glasgow, U.K.

The CATO research project started out in 1983 as an endeavour to produce software incorporated into three topics, one for each stage of primary, to test the theory that the introduction of technical innovation in primary teaching will only succeed with most teachers if well-embedded in the best of normally accepted good practice. The idea was to link COLLEGE LECTURERS EXPERIENCED IN PROVIDING INSERVICE COURSES WITH TEACHERS EXPERIENCED IN A SUCCESSFUL TEACHING APPROACH TOGETHER WITH A PROGRAMMER TO FORM A RESEARCH AND DEVELOPMENT TEAM.

In the CATO project we have had research funds allocated over a five year span to a project which reflected the needs of practitioners and the results of that research have been delivered in a readily accessible form to the practitioners, namely computer assisted topic packages for each stage of the primary school.

DESPERATE JOURNEY - Scottish Highlanders to Kildonan, Canada in 19th Century
TRAVELLING SHOPS - Designing an urban estate without shopping facilities
FANTASY ISLANDS - Mathematical islands inhabited by interesting families
WHALES - Planet earth, man and the whale down the centuries
ANIMAL KINGDOM - Creatures created for desert, grassland or forest uplands

Each package has a hefty full colour teaching manual and between 15 and 25 computer programs ranging across the whole spectrum of software types.

EMAO - ENSEIGNEMENT DE MATIÈRES ASSISTÉ PAR ORDINATEUR

Le projet EMAO, commencé en 1983, avait pour but de produire des logiciels intégrés à trois matières, une pour chaque étape du niveau primaire, en vue de vérifier la théorie selon laquelle l'introduction d'une innovation technique d'enseignement primaire ne peut réussir avec la plupart des enseignants que si elle est bien incorporée à la meilleure des pratiques courantes. L'idée était de réunir DES CHARGÉS DE COURS, AU NIVEAU COLLÉGIAL, HABITUÉS À DONNER DES COURS INTERNES, DES PROFESSEURS AYANT UNE MÉTHODE D'ENSEIGNEMENT EFFICACE ET UN PROGRAMMEUR EN VUE DE FORMER UNE ÉQUIPE DE RECHERCHE ET DE DÉVELOPPEMENT.

Dans le cadre du projet EMAO, des fonds de recherche ont été attribués, durant une période de cinq ans, à un projet qui reflétait les besoins des praticiens; les résultats de cette recherche ont été communiqués, sous une forme directement accessible, aux praticiens, notamment des didacticiels pour chaque étape de l'école primaire.

DESPERATE JOURNEY - Highlanders écossais, à Kildonan, au Canada, au 19e siècle
TRAVELLING SHOPS - Conception d'un domaine urbain sans installations commerciales
FANTASY ISLANDS - Îles mathématiques habitées par des familles intéressantes
WHALES - La planète terre, l'homme et la baleine au cours des siècles
ANIMAL KINGDOM - Créatures créées pour le désert, la prairie ou les régions montagneuses et forestières)

Chaque progiciel comprend un gros manuel d'enseignement en couleur et de 15 à 25 programmes couvrant tout le spectre des types de logiciels.

CATO - COMPUTER ASSISTED TOPICS

Patricia R. Watterson

Jordanhill College of Education, Scotland

The CATO research project started out in 1983 to test the theory that the introduction of technical innovation in primary teaching will only succeed with most teachers if well-embedded in the best of normally accepted good practice. The idea was to link

LECTURERS EXPERIENCED IN PROVIDING INSERVICE COURSES IN THE TOPIC METHOD
WITH TEACHERS EXPERIENCED IN USING THE TOPIC TEACHING APPROACH
TOGETHER WITH A PROGRAMMER
TO FORM A RESEARCH AND DEVELOPMENT TEAM.

The particular circumstance which led to the research project was the quite sudden provision in 1983/84 of half price microcomputers for every primary school in an effort by the Department of Trade and Industry to galvanise British educators into participating in the revolution affecting society; the rapid introduction of micro-technologies. Unlike, for example, Scandinavia where schools were flooded with gifts of hardware left over from businesses marching forward rapidly and outgrowing hardware systems, the British offer was not made to assist hardware houses produce the next generation of hardware. British schools were offered systems designed specifically for educational use at a time when schools just did not know what kind of educational use they could make of any system.

The aim of all inservice education is to update teacher skills in the light of current research and changes in the needs of society. How did the provision of microcomputers alter this basic premise? The sudden unexpected resourcing of schools necessarily skewed the direction of types of inservice courses offered to teachers. Education Authorities anticipated a clamour for courses in microcomputing. The problem, of course, lay in the fact that there was no software of any great educational worth yet developed. The only teaching personnel within Education Authorities with experience in the use of computers were in Maths or Physics Departments, areas of teacher scarcity where teaching styles had little in common with current primary classroom methods. One or two lecturers in each College of Education in Scotland had become involved in pilot schemes to look at probable computer use and the national body, SMDP, was funded to work with teacher groups to develop software. In all of these initiatives, the microcomputer was the starting point.

The basic premise of the Government's offer of hardware was that the provision of the resource would inevitably initiate change. A Government department which was not the Education Department decided that it could provide the only catalyst that was wanting. However, educationists knew that the teachers themselves would have to be convinced that change was necessary and that teachers would have to become part of the change process if any educational use were to be found for the gifts of hardware.

The action research solution.

The core of the CATO team consisted of lecturers who had been working with primary teachers for many years developing a method of inservice provision that helped the teachers with the demands of society for flexible learners BY GIVING PRACTICAL EXPERIENCE OF MORE OPEN LEARNING. Topic teaching MATERIALS WHICH THE LECTURERS HAD EVOLVED AND DEVELOPED WITH TEACHERS AND THEIR CLASSES were becoming major curriculum resources, though in 1983 none of the course materials had been published. The starting point for the CATO project was very different from other projects. It was to work with experienced teachers at each of the main stages of the primary school, to research together known practice IN THE CONTEXT OF A SUCCESSFUL TOPIC and find the areas where a teacher needed further support to fulfil the learning objectives. The

research question was whether computer software could be developed to improve the implementation of the topic method. The TOPIC APPROACH (1) upon which the CATO project is based is a teaching method using collaborative story telling, one of the oldest and most enduring forms of human education. Through this method children are carried in an upward spiral of problem tackling as the storyline of the topic unfolds. For the purposes of the research three very different styles of topic were chosen, one for each stage of the primary school. The first, DESPERATE JOURNEY, was based on the history of a Scottish family burned out of their home during the land clearances, forced to find work in the big city early in the industrial revolution, and eventually emigrating to the new lands of Canada. The topic was designed for children aged 9 to 12 years and developed class work across the curriculum. The second topic for 7 to 9 year old children was TRAVELLING SHOPS; the storyline led the children to develop an open site into a modern housing development on which the planners omitted provision for shopping. The children invent the solution of shopping vans which they create and operate as an art/craft/maths/language/ technology/drama series of activities. The third topic, for children aged 5 to 8 years, was FANTASY ISLANDS which has an underlying mathematical pattern to it .

For each of these very different topics, designed to support the distinct stages of child development in the primary years, the teaching method was basically the same. Software and computer peripherals were designed to support the teacher needs and the children's learning; resulting in packages which, over the span of the five years, incorporated the whole range of computer experience in a user friendly way. Without trauma, primary teachers were making extensive use of databases, word processing, simulation, control technology before being aware of the computer terminology for such activities. The learning needs pressed the computer developers to find a solution: the computer program did not come begging for a place in the scheme of things. This was the main difference between the CATO project and other inservice computer developments.

The teacher as action researcher.

The teachers, in the beginning, did not know what software was or how microcomputers worked. The teachers DID KNOW where they needed support to achieve aims. Some of the support they sought was very predictable: books; well tried and tested teaching notes; examples of other teachers' successful classroom outcomes; a good topic outline; ready access to relevant information within television programmes; linked video material; and time together to discuss methods and ways round failed methods. Very quickly within the project they perceived how attentive children were to interactive computer programs related to ongoing class work. Holding interest and attention are first level requisites in teaching. Very quickly the teachers noticed that the interaction was variable; being either between children and the microcomputer or, even richer, including the teacher in the interaction as another teacher or as a fellow learner. Better still, the interaction of children in a group situation with a microcomputer meant that the teacher had an opportunity for objective analysis and assessment of learning which was closed to her in the traditional class setting where she had to attempt to teach and assess at the same time. The assessment picture had broadened for the teacher and the teaching activity, consequently, became more interesting and slightly less stressful. Now the teacher was warming to the research role and wanted software designed to tackle persistent learning difficulties. Bigger questions could then be framed. Was there a learning difficulty or a mismatch in task? Was there a need for children to tackle this activity at all? How had it crept into the curriculum in the first place?...
The process continues.

The place of the Inservice course in the research process.

Although the core of the development work took place in the 12 pilot schools and was shaped by the participating classes, another important element came from inservice courses, mostly, of three days' duration, where a wider range of teachers experienced the method. Local Authorities made use of government initatives to enable the teacher on such courses to purchase the packages as they were published. The teachers were attending the courses because they wanted to learn how to use computers with their classes and to broaden their experience of the topic approach. On such courses further important refinements were made to the software to accommodate the computer inexperience of most teachers. Also, the teachers' notes in the packages were refined as a result of feedback from teachers on

inservice courses. The research assistants attached to the project team were all primary teachers, each in turn seconded to the project and funded by the local authority. The research assistants worked closely with the lecturers and the programmer gaining and sharing skills so that on return to their posts they could become part of the local support team for primary teachers and computer education.

SINCE 1983, THROUGH THE CATO PROJECT, JORDANHILL'S REPUTATION FOR DEVELOPING SOFTWARE WITH WIDE TEACHER RESOURCES LINKED TO GOOD INSERVICE COURSES HAS GROWN AND BEEN SEEN TO SUPPORT THE FASTEST GROWING AREA OF INSERVICE TEACHER EDUCATION. The network developed with local authority inservice providers has been unique in a college system such as Scotland's where the college and school systems have no administrative connection. THE STRENGTH OF THE CATO DEVELOPMENT HAS BEEN ITS IMMEDIATE LINKS WITH INSERVICE PROVISION. The immediacy of the two-way process through inservice courses and school focused development prevented the college lecturers developing 'ivory tower' syndrome or retreating into the mystic superiority of computer jargon. The programmer was unable to distance himself from the clients, teachers and children, or the developers. The ability of every member of the small team to continually adjust to the changing needs in such a fast growing area meant that the outcome was always appropriate. Software not linked in the way that the CATO packages have been to cross curricular work has enjoyed a brief popularity and then lain unused.

THE LONGER LIFE OF THE CATO PACKAGES CAN BE ATTRIBUTED TO THE INCORPORATION OF THE SOFTWARE IN A FULLY RESOURCED TEACHING PACKAGE BASED ON A TEACHING APPROACH THAT HAS DEVELOPED WITH TEACHERS THROUGH INSERVICE WORK OVER A NUMBER OF YEARS. THE COMPUTER HAS NOT BEEN THE PRIME FOCUS: THE LEARNING PROCESS HAS BEEN THE CENTRAL ISSUE.

For the CATO team there has been much more gained than the obvious satisfaction of producing a computer assisted resource with a reasonable educational life span built into it. There has been a strong feeling of continual personal development and mental stimulation for everyone involved in the venture. The records kept of distribution and inservice demand by the Regional Authority Computing Advisory teams, together with individual purchase records held by the college, indicate that practically every primary school in Scotland in possession of a BBC microcomputer system is using the CATO packages.

In the 1987 SERA lecture (2) Ralph Wilson put forward his view that the three key principles in the management of change in education are as follows:-

1. Educational decision-making should be firmly based on some explicit statement of educational philosophy.

2. The management of educational change should seek to liberate and extend the professionality of the educators on whom change depends.

3. If an educational development is to succeed, the participants must feel a sense of ownership of and belonging to the development.

Against such criteria the CATO team have measured the success of their development and not found it wanting. We have found a way of working with classroom teachers as co-researchers in this endeavour based on the topic approach to teaching. We firmly believe that the topic approach we have developed succeeds because it offers teachers and children a means of starting always from what is known within each child's own understanding rather than some arbitrary point in a curriculum chosen for likely best fit to the greater number in the class. The approach is not one of 'problem solving' but rather one of common sense 'problem tackling' within a common context. So, too, the approach is ideal for teachers navigating their way in the age of the information explosion where success will depend on efficient use of the new technologies. The CATO packages offer a full resource to enable the classroom teacher to assimilate the new technologies with confidence.

In the CATO project we have had research funds allocated over a five year span to a project which reflected the needs of practitioners and the results of that research have been delivered in a readily accessible form to the practitioners, namely computer assisted topic packages for each stage of the primary school.

The Computer Assisted Topic Packages

DESPERATE JOURNEY
(Scottish Highlanders to Kildonan, Canada in 19th Century)
TRAVELLING SHOPS
(Designing an urban estate without shopping facilities)
FANTASY ISLANDS
(Mathematical islands inhabited by interesting families)
WHALES
(Planet earth, man and the whale down the centuries)
ANIMAL KINGDOM
(Creatures created for desert, grassland or forest uplands)
Each package has a hefty full colour teaching manual and between 15 and 25 computer programs ranging across the whole spectrum of software types. All the packages are published by Jordanhill College Sales & Publications Unit, Glasgow, Scotland.

1. Rendell, F.W. The Topic Method: How and Why . Jordanhill College, 1985.
2. Wilson, R. The SERA Lecture, 1987. Scottish Educational Review, Vol 19, No21, Edinburgh.

SERA Scottish Educational Research Association
CATO Computer Assisted Topics

THE USE OF COMPUTER ASSISTED LEARNING IN DISTANCE EDUCATION

S.E. Phillips
North Island College
Comox, BC

Computer Assisted Learning (CAL) has only recently been utilized to any extent in the area of Distance Education. There are a variety of reasons that CAL has not been used more, one of the most obvious being the absence of good, usable, timely and appropriate CAL courseware.

Since the inception of North Island College in 1975, we have been dedicated to supplying education to our students in their home communities. This means providing distance education through a variety of methodologies, one of which is CAL.

To this end, in the fall of 1985, a group was formed at North Island College to produce CAL courseware that could be used throughout the college region. Since that time, the Computer Options for Learning and Training (COLT) group has been creating courseware, primarily within computer-related fields. It has also been testing and developing new tools for efficient CAL courseware production.

This paper discusses the advantages and limitations of using CAL for distance education and the necessity of a team approach throughout the process of CAL courseware creation.

L'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR DANS LA FORMATION À DISTANCE

C'est depuis peu seulement que l'enseignement assisté par ordinateur (EAO) est utilisé, dans une certaine mesure, dans le domaine de la formation à distance. Plusieurs raisons peuvent expliquer cette utilisation restreinte, l'une des plus évidentes étant l'absence de bons didacticiels EAO.

Depuis l'ouverture du North Island College en 1975, nous nous sommes appliqués à dispenser l'enseignement à nos étudiants dans leurs communautés respectives, c'est-à-dire que nous leur offrons une formation à distance au moyen de diverses méthodologies, dont l'EAO.

À cette fin, un groupe a été formé au North Island College à l'automne 1985 pour élaborer un didacticiel EAO qui pourrait être utilisé dans toute la région avoisinante. Depuis ce temps, le groupe Computer Options for Learning and Training (COLT) crée des didacticiels, ayant essentiellement trait aux domaines para-informatiques. De plus, il teste et met au point de nouveaux outils pour améliorer la production de didacticiels EAO.

Dans cette communication, nous discuterons les avantages et les inconvénients de l'EAO pour la formation à distance et la nécessité d'un travail d'équipe durant tout le processus de création de didacticiels EAO.

THE USE OF COMPUTER ASSISTED LEARNING IN DISTANCE EDUCATION

SUSAN E. PHILLIPS

NORTH ISLAND COLLEGE
156 MANOR DRIVE
COMOX, B.C.
V0N 6P7

Introduction

The North Island College region is situated in a rugged area of British Columbia. The region extends from mid-Vancouver Island north, and along the central British Columbian mainland coast. It covers approximately 80,000 square kilometres, much of which is heavily forested and mountainous. Logging roads are the only means of access to many of the communities (other than by seaplane or boat for those on the coast), and are often impassable a good part of the year.

Although much of the North Island College region is remote and difficult to access, 80 percent of the region's estimated population of 131,000 live in one of the three main communities: Port Alberni, Campbell River and the Comox Valley. There are major North Island College centres at these sites, and smaller centres in many of the less populated communities such as Port Hardy, Gold River and Alert Bay.

The College's mission states that the College is "...committed to providing educational services to members of the public in the communities where they live and in the circumstances and conditions that currently prevail for those persons in their communities." This mission requires that not only students within easy access of the main centres, but all students within the region, must have access to the courses that North Island College provides. In order to satisfy the mission, teaching methodologies had to be found and implemented that would work both at the remote sites (without subject instructors present) and at the larger learning centres. Computer Assisted Learning (CAL) was investigated and eventually adopted as one of the means to provide flexible, self-paced instruction under a variety of circumstances.

When plans for North Island College were being developed, the Regional Advisory Committee envisioned it as an institution that would provide education to"... traditional clientele with traditional and non-traditional delivery systems." Furthermore, North Island College would "... take the college to the community". After investigation of various instructional methodologies, CAL seemed to be an approach that would work for the delivery of education to students in remote areas as well as in the more densely-populated regions.

One of the most important features of CAL for North Island College is that it permits students to take courses in their home towns, and does not necessitate their having to relocate or to travel in order to receive the education they desire. It is not feasible for the College to employ specialized instructors for all small centres and CAL allows all students throughout the region to receive the same instruction, written by subject experts, within their own locale.

CAL has other advantages which are present whether a course is offered in a remote village or in a large urban centre. For example, computers are infinitely patient. If a student does not understand a particular concept, he can repeat the on-line section over and over again. Similarly, if a student needs to review, he can quickly browse through earlier parts of the CAL. The computer does not care how many times a student repeats certain sections; it just does what it is told. Students can repeat tests and exercises until they have mastered the concepts and achieved the instructional objectives.

A third advantage of CAL is that the on-line courseware basically replaces the lecture in the traditional methodology of teaching. This means that as long as a student has access to the CAL and to a compatible microcomputer, he can take the courseware when it is convenient for him. The "lectures" do not just take place at a certain time each week; they can be taken on convenient days, or omitted one week and doubled up another if the student so chooses. This leads to great flexibility and accommodation for the student, and is especially important when dealing with mature students who may have many other commitments, such as families and jobs.

The Limitations of CAL

CAL adversaries argue that the quality of instruction may not be as good when delivered on a computer as compared to delivery in the traditional mode. This argument is usually based upon the unsubstantiated conjectures that all instructors are good teachers and that all CAL is poorly-developed. Generalizations like these are not supported by research. There is good CAL, just as there are poor instructors, and a teaching institution must evaluate the instructional methods, available materials and feasible costs for the subject areas, and make decisions based upon these evaluations.

A second argument against CAL is that it elimates the very important interpersonal relationships between the student and the teacher and among the students themselves. However, the number of students in any subject area in remote communities of the College region is small — not even approaching the number needed for a normal class size. Therefore, student-to-student interaction in these subject areas would not be present in any case unless the student did move to a larger centre, thereby contradicting the mission of the College.

When using CAL, the instructor and students, although not in close physical proximity, certainly do have a relationship. Each student communicates with a subject matter instructor via telephone lines or computer communication networks. This type of communication may be of higher quality than that of the instructor-to-student in large first-year classes at traditional universities and colleges. For example, one advantage of communicating electronically is that a student receives very rapid feedback, often within a few hours, rather than having to make an appointment and wait until it is convenient for both parties to meet.

A very real drawback to CAL is the development time that CAL courseware requires. Estimates vary, but the average time is somewhere between forty and one hundred hours of development time for each hour of instruction. This, of course, is highly dependent upon the subject of the instruction and the quality of the work being produced. However, regardless of the topic, and even of the quality of the courseware produced, CAL production is expensive both in terms of money and of time commitment.

The long development time leads directly to another limitation, that of updating the CAL courseware. When a course has taken a long time to develop, it probably will not be rewritten, or even updated, for several years. The problem may exist where a course in a subject area that is rapidly changing may be outdated even before it is released.

However, the long development time is needed to produce the high-quality CAL courseware; the CAL would not be used if the quality was not there. Furthermore, without CAL, students in remote areas would not be receiving the courses that they want, as the developmental costs are still considerably lower than having instructors available in small centres, or relocating the students.

The Team Approach to Developing CAL

In the view of the administration of North Island College, it is clear that the advantages of CAL outweigh the limitations, at least in the College region, and therefore the decision was made to continue to produce and support CAL courseware.

The team approach has been the traditional approach for CAL creation ever since CAL became an accepted mode of instruction. With the development of sophisticated authoring systems, some vendors are claiming that this approach is outdated and no longer required.

However, at North Island College, we have found that the team approach is the only approach that will reliably produce usable CAL.

The team members for a specific project will vary depending upon the subject of the courseware, the format the courseware is to take, the supplementary materials that are to be included, and the software that is to be utilized for that particular piece of CAL. However, at the College we have found that a team composed of individuals in the following positions works best:

1. Project Manager

Even if only one person is working on a project, it is a good idea to have another person acting as a project manager. This will help to ensure that development is on schedule and that deadlines are being met. When there is more than one person on a project, the manager also takes on the role of liasing between the individuals involved.

2. Subject Matter Expert

It is not reasonable to expect that a CAL development team will have the necessary expertise in any given area to write a course. Even with a number of reference books, it is essential that a person who is an "expert" in the area be available for consultation, if not for the actual content writing. It is also important that the same expert review the final courseware, checking the accuracy of the material, as editing changes may inadvertently cause inaccuracies in the information presented.

At North Island College, the subject experts are often only part-time members of the team. They may be staff members with other duties, or they may be from outside the College region contracted to supply the expertise and content material.

3. Programmer

Depending upon the authoring systems, languages and other software that the team is using, a programmer may or may not be needed on the team. However, at the College, we have found that our team could not continue to use the software that is available, or follow the approaches that are taken, without employing at least one programmer on a full-time basis.

4. Technical Writers/Editors

Writers and editors are very important to the team as they are responsible for editing the information that the subject matter experts submit. Often, they must add to or enhance the material in order to produce a course that flows smoothly and logically. They are usually responsible for the formatting, as well as the editing, of both the on-line material and the hardcopy student manuals.

5. Instructional Designer

The instructional designer must work closely with the subject matter experts in determining the format and design of the courseware. Often the subject expert will have had little or no experience with CAL and it is up to the designer to work with the expert in creating a feasible plan for the on-line material.

Depending upon the expertise and experience of the other team members, the instructional designer must also work closely with the programmer and writers to ensure that the subject expert's directions are being interpreted correctly and that good design principles are being followed. The instructional designer is often responsible for the final product that is released and therefore must ensure that the on-line material and the supporting print documents are all complete and of professional quality.

6. Graphic Artist

Graphics are often a key component of CAL courseware — the quality of the graphics may make the difference between an often-used course and one that is seldom touched by either the students or the instructors. However, some courseware has no need for any graphics, and in others, the graphics can be included within the courseware through the use of a scanner or clip art picture collections. It is, therefore, fairly common for graphic artists to be hired on contract for a short period of time for a finite amount of work. At the College, we do not have a graphic artist as a permanent member of the team, although we have been fortunate in that

people hired for other positions have been able to create the necessary graphics.

7. Beta-testers

It is critical that all CAL courseware be carefully beta-tested before it is released to students. This is an important step for any course being produced, but with CAL there is the added potential problem of computer-related obstacles occurring. Directions must be very clear and there must be no assumptions made about the student's ability to "figure out" how something works.

Often a team member can beta-test the courseware, although it is advisable to have someone not directly connected to the team do the testing. In this way, the tester will not be influenced by knowing and working with the course developers, or by the knowledge of the process used to develop the course.

At North Island College, we try to use instructors or senior students out in the field to do the actual beta-testing. This step is carried out after a very thorough testing has been completed within the team itself.

Summary of the Team

Although a number of different positions in a CAL development team have been identified, it is not uncommon for one person to fulfil more than one of the positions. In our development team, the instructional designer often doubles as a manager, and the writers often write and/or test the courseware. This is reasonable, considering that the team fluctuates in size, in the amount of time available to spend on a given project, and in the individuals' background knowledge and skills. In order to keep team members occupied at all times, and to get projects out by deadlines, it is essential that no one person on the team insists upon doing one task only. Flexibility amongst the team members is vital, as is a willingness to learn and attempt tasks outside a person's given expertise.

Other Considerations When Using CAL

An implicit requirement of implementing CAL is to have computers available for the students to use. This requires a large up-front investment for hardware and software. It also necessitates research into the market as to what is available and to the requirements of the particular CAL, in terms of operating systems, processor speeds, display capabilities, add-on peripherals, internal memory and disk drives.

A second consideration is that the students must be shown how to access the CAL. This means that personnel must be available to demonstrate the required procedures and to do limited trouble-shooting. It is very frustrating for a student to try to work through a CAL lesson, only to have the computer or the diskettes work incorrectly. CAL developers should strive to make the computer as transparent as possible to the student, particularly when the subject matter is non-computer related.

Conclusions

The use of CAL courseware in the College region has proven to be a viable alternative to traditional modes of instruction. It meets the need for open, self-paced courses, and reduces the number of subject instructors needed. It also provides College courses to all members of the population within the region, regardless of where they live or the amount of time they have to dedicate to their studies. CAL has proven to be successful and effective for the North Island College population.

References

Regional Advisory Committee Report, North Island and Powell River Community College, March, 1975.

Won, T.K., Using Computer Based Training (CBT) in an Open Learning Environment, May, 1986.

INTERNATIONAL COOPERATION IN CAL

R.D. Monaghan, G. Welch
Cambridge Language Arts Software Services (CLASS) Inc.
Montreal, PQ

L.R. Winer
Centre québécois de recherche sur les applications pédagogiques de l'ordinateur (APO Québec)
Montréal, PQ

The transfer of educational materials from Europe to North America raises some fundamental concerns about the very nature of CAL. For this transfer to be effective, it must be based on a shared philosophy of education, a recognition of official guidelines, and an appreciation of the politics of classroom implementation. An effective transfer may require more than just translation, or even adaptation; often a true redevelopment process must be undertaken. The process involved in redeveloping British language arts software, originally developed at the University of Cambridge, to meet classroom needs of teachers in Québec will be reviewed and illustrated through discussion of a content-free language exploration package (READER/WRITER), a whole language simulation package (SPACE PROGRAM ALPHA), and an affective gaming package that addresses the subtext of a novel (BIG BROTHER). Particular attention will be paid to design considerations and to the formative evaluation process to which all the packages were subjected. Projects such as the one discussed here, a joint venture between APO Québec and CLASS (the Canadian link to CLASS UK), indicate the potential for much greater international cooperation in CAL.

COOPÉRATION INTERNATIONALE DANS L'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR DU LANGAGE ARTS

Le transfert de matériels pédagogiques entre l'Europe et l'Amérique du Nord soulève des questions de fond sur la nature intime de l'EAO. Pour qu'un tel transfert réussisse, il doit s'appuyer sur une philosophie commune de l'enseignement, une reconnaissance des directives officielles et une évaluation de la politique d'implantation dans une classe. Un transfert réussi suppose souvent plus qu'une simple traduction, voire adaptation; parfois, il faut envisager une véritable refonte. C'est le processus de refonte du logiciel britannique du langage arts, mis au point à l'université de Cambridge, en vue de l'ajuster aux besoins des enseignants du Québec que nous relatons et illustrons ici. Pour ce faire nous analysons un progiciel d'exploration de langage à contenu libre (READER/WRITER), un progiciel de simulation de langage complet (SPACE PROGRAM ALPHA) et un progiciel de jeu affectif qui traite le sous-texte d'un roman (BIG BROTHER). Nous nous intéressons surtout aux considérations conceptuelles et au processus d'évaluation de la formation auquel on a soumis ces progiciels. Les projets comme ceux décrits ici, coentreprise entre l'APO Québec et CLASS (le représentant canadien de CLASS UK), laissent entrevoir les avantages d'une plus grande collaboration entre les pays sur l'EAO.

INTERNATIONAL COOPERATION IN CAL

Richard D. Monaghan
Cambridge Language Arts Software Services (CLASS) Inc.
Montréal

Graeme Welch
Cambridge Language Arts Software Services (CLASS) Inc.
Montréal

Laura R. Winer
Centre québécois de recherche sur
les applications pédagogiques de l'ordinateur
(APO Québec)
Montréal

In recent years, there has been an increasing emphasis placed on using computers to model environments for learning rather than environments for teaching. Nowhere is this more evident than in the teaching and learning of the language arts. Over the last decade and more, the inclination of Canadian and British educators has been towards a model of learning that encourages exploration of language through social interaction, and a recognition that the learning of language is not a lock-step procedure through stages of reading and stages of writing (Barnes, 1975; Robinson, 1985; Smith, 1986, 1988; Newman, 1985). The language skills acquired by students are not disparate compartmentalized and identifiable behaviours. There is a complex interplay between reading, writing, listening, speaking and interpreting visual signs in terms of our own experience that determines our ability to manipulate language with confidence and grace. To be effective, computer-assisted-learning materials for the language arts must recognize this reality.

Content-free software has done much to enhance the development of learning skills in a wide range of subject areas. Word processing and drawing packages, as well as music construction kits, have provided students with learning environments within which they can explore and develop aptitudes and strategies for making and sharing meaning (Snyder & Palmer, 1986; Daiute, 1985). More and more powerful machines, beyond the reach of most schools, have prepared the way for educational packages that encourage experiential learning. At the same time, we cannot ignore the potential for developing new kinds of software for the installed base in Canadian schools.

The project under discussion here is the development of content-free language arts software for English mother-tongue and French mother-tongue students in Québec schools. The computer programs and related materials have been designed specifically to address the curricular guidelines established by the Ministry of Education of the Province of Québec for language arts teaching and learning; that is, they have been designed to foster a student-based whole-language approach to the learning of language (Ministère de l'Education, 1982, 1983). Originally designed in England, the programs were redeveloped in Canada as a joint venture between Cambridge Language Arts Software Services (CLASS) Inc. and le Centre québécois de recherche sur les applications pédagogiques de l'ordinateur (APO Québec).

The whole-language approach to the language arts curriculum has proponents in Britain, Canada, Australia, New Zealand and the United States. The bibliographies in current literature demonstrate just how cross-cultural the theory and practise is (Newman, 1985). The Ministry of Education of the Province of Québec has adopted a whole-language model in its new English mother-tongue language arts curriculum, and the French mother-tongue language arts curriculum stresses oral as well as writing and reading aspects of language development. The similarity in the pedagogical goals of certain British educational materials to the stated goals of the new curriculum in Quebec encouraged us to examine the potential of the specific computer-based learning materials in Québec schools.

The programs described below were originally designed by a team of educators at the Department of Education at the University of Cambridge in England. Rights to experiment with program design and implementation in North America were acquired from Cambridge Language Arts Software Services Ltd. by Cambridge Language Arts Software Services (CLASS) Inc. The original program design was intended to provide students with a means to explore dimensions of their own language through the creation of text and graphics screens; they should then be free to encrypt their own creations to reconstitute or their own or as challenges to others. Essentially, the programs allow children to compose text and graphics pages, save them, and then manipulate the pages in the following ways: alphanumeric characters are replaced with asterisks and students reconstitute the original screen presentation

(meaning) by replacing the asterisks with text; textstrings are jumbled on the order of characters in individual words, words in sentences, sentences in paragraphs, or lines on the page – the object is to reconstitute the intended meaning of the original; pages are ordered in sequences that allow the creation of databases or branching stories – the object is to encourage students to "track" information or to examine narrative structures. in more general terms, the programs are intended to provide activities that encourage students to hypothesize solutions, predict outcomes, and to verify their hypotheses. The program design actively encourages students to work together to create puzzles for each other with their own language, in their own registers. Five other exploration packages were included in this series, but they are outside the range of this presentation. The electronic medium for this combined textual and graphic representation of meaning was the alphamosaic videotext system of the BBC microcomputer.

At the same time, apart from this content-free set of computer-based-learning-materials, four other packages were designed in Britain to be used by students working in groups; the object of these computer programs and printed materials was to provide a stimulating context for collaborative learning. Two packages provide learning modules for space travel simulations, and two provide simulations leading up to and away from a study of Orwell's 1984. What distinguishes the space travel simulations from others is the extensive recommended range of language activities to be undertaken away from the computer; these activities include report writing, letter writing, taping log entries on cassette, designing passports, and describing new worlds and old. The Orwell packages also include extensive language-related activity away from the computer (including drama activities). The programs for these four units were written for the videotext display of the BBC microcomputer.

The redevelopment of the programs was not intended to be merely an adaptation from one machine to another, or from one language to another. There is really little point in deliberately building the constraints of one machine into another machine that is free from those constraints. The programs were redesigned from scratch to meet not only pedagogical concerns, but the dominant machine base of Québec schools. It was decided to redevelop four of the content-free units and one simulation package.

The design team consisted of a project manager, program analyst, and pedagogical consultant. The pedagogical goals were established first, adhering to the publications of the Ministry of Education of the Province of Québec and confirmed through consultations with Ministry officials. Functional specifications of the computer programs followed. For example, the programs were expected to permit students to enter, delete, copy and move blocks of text, to enter graphics, to encode or decode textstrings, and to manipulate the display of text through foreground and background colour and size. The goals these specifications were expected to meet were, through writing, to provide writers with tools to specify meaning (through different uses of emphasis and the creation of contexts for expression), and through reading, to provide readers with activities that encouraged pattern recognition and hypothesis-testing. These steps in the design of the programs were subjected to regular formative evaluation through the services of APO Québec (Bergeron, Carrière, Lafeuille & Roy, 1987). It was decided early in the project to apply the principles of evolutionary software design and to implement regular formative evaluation on all stages of the design, encoding and testing of the products (Gilb, 1988). The evaluation instrument was a technical implementation grid listing pedagogical goals against functional specifications.

The machine base in Québec is overwhelmingly MS-DOS compatible. Aside from the Max20, developed specifically for Québec schools, PC's and PC-compatible machines are installed in most primary and secondary schools. Redevelopment for the MS-DOS machines assumed a minimal configuration of 256K, double disk drive, and a CGA adaptor with a colour monitor. Apart from smaller concentrations of Icons and other machines, the second most dominant machine in Québec schools is the Apple. The Apple version of the programs assumed an Apple IIGS with 512K, dual 3.5" or 5.25" drives, and RGB monitor. The ImageWriterII printer for the Apple IIGS has the capability of printing in colour, so we included that possibility in the program specifications. Earlier research had indicated that the other Apple II models could not support the intensive use of graphics demanded by videotext presentation.

The formative evaluation process was extremely effective in allowing us to recognize potential problems. Significant difficulties were encountered in the Apple IIGS versions of the programs. In January it became apparent that, for considerations of time and expense, the Apple IIGS implementation of the programs would have to be curtailed.

Once beta versions of the programs and documentation were ready, they were introduced into schools where students and teachers continue the formative evaluation process. The use of the programs is being measured by an evaluative grid very similar in design to the technical implementation grid mentioned above, by recorded observations of student and teacher use, and by interviews with teachers and students engaged in using the programs. Simultaneously, the use of the programs is being monitored by researchers at APO Québec's *Vitrine 2001* (Winer, 1988). As suggestions for improvements to the programs, documentation or implementation arrive, they are considered for inclusion into the prototypes.

The redeveloped packages will be tested again in British schools over the next semester, on MS-DOS compatible machines. We hope soon to have students from both countries exchanging files.

The main goal of this project was to determine if Ministry guidelines for student-centred whole-language learning could be effectively complemented in our schools by computer-assisted-learning materials originally designed elsewhere for whole-language learning. Preliminary results indicate that by adhering to specified pedagogical goals from the outset, such a transfer of pedagogy and technology can be effected. We have discovered, in the process, that a most efficient way of effecting such a transfer is through an evolutionary approach to project design and implementation.

REFERENCES

Barnes, D. (1976). *Communication to Curriculum.* Harmondsworth: Penguin.

Bergeron, G., Carrière, M., Lafeuille, J., et Roy, S. (1987) *Un modèle de recherche-developpement en APO.* Montréal: APO Québec.

Daiute, C. (1985). *Writing and Computers.* Reading, Mass.: Addison-Wesley.

Gilb, T. (1988). *Principles of Software Engineering Management.* Wokingham, U.K.: Addison-Wesley.

Ministère de l'Éducation, Gouvernement du Québec. (1982). *Guide pédagogique: Français: première, deuxième, troisième années.* Québec: Gouvernement du Québec.

Ministère de l'Éducation, Gouvernement du Québec. (1982). *Guide pédagogique: Français: quatrième, cinquième sixième années.* Québec: Gouvernement du Québec.

Ministère de l'Éducation, Gouvernement du Québec. (1983). *English Language Arts I-VI.* Québec: Gouvernement du Québec.

Ministère de l'Éducation, Gouvernement du Québec. (1983). *English Language Arts: Language: Secondary School Curriculum Guide.* Québec: Gouvernement du Québec.

Newman, J. (Ed.) (1985). *Whole Language: Theory in Use.* Portsmouth, N.H.: Heinemann.

Robinson, B. (1985). *Microcomputers and the Language Arts.* U.K.: Open University.

Smith, F. (1986). *Insult to Intelligence: The Bureaucratic Invasion of Our Classrooms.* Portsmouth, N.H.: Heinemann.

Smith, F. (1988). *Joining the Literacy Club.* Portsmouth, N.H.: Heinemann.

Snyder, T. & Palmer, J. (1986). *In Search of the Most Amazing Thing.* Reading, Mass.: Addison-Wesley.

Winer, L.R. (1988). Research on computer-based learning environments: The Vitrine 2001. *CJEC* (17:3), Fall 1988.

COMMENT DES ENSEIGNANTS DU PRIMAIRE S'APPROPRIENT UN LOGICIEL ÉDUCATIF

H. Kayler
Maîtrise en enseignement au primaire
Université du Québec à Montréal
Montréal, PQ

Nous élaborons actuellement, grâce à une subvention de APO/Québec, un EDIPO (Ensemble Didactique d'Intégration Pédagogique de l'Ordinateur) que nous expérimentons auprès d'enseignants du primaire.

Ce matériel est construit à partir de nos conceptions de l'enseignement des mathématiques au primaire et de l'utilisation de l'ordinateur. Ces conceptions reposent sur nos expériences tant en recherche (sur l'apprentissage des mathématiques et sur l'apprentissage de la pensée procédurale grâce à une subvention FCAR) qu'en enseignement en formation et perfectionnement des maîtres en mathématiques (cours de didactique à l'Université). La démarche pedagogique et le contenu mathématique proposés sont en relation avec les orientations du programme de mathématiques du Ministère et l'on y fait place, en particulier à: l'importantce de la géométrie, l'importance de la résolution de problèmes, l'approche constructiviste de l'apprentissage, la démistification de l'erreur dans le développement des connaissances, l'importance des liens entre les différentes disciplines mathématiques.

Ce matériel prend la forme d'une suite d'explorations et de constructions de concepts mathématiques; ce sont des activités où l'enfant travaille tantôt sur l'ordinateur, tantôt sur papier (à l'aide de crayon, règle, rapporteur et compas). L'accent est mis soit sur des activités individuelles ou en équipe, soit sur des explications de l'enmseignant, soit sur des discussions collectives.

Une présentation détaillée de ce projet, ainsi qu'une première analyse des réactions des enseignants et des élèves fera l'objet de la communication proposée.

HOW PRIMARY SCHOOL TEACHERS ACQUIRE LEARNING SOFTWARE

With the help of an APO/Québec grant, we are currently developing an EDIPO (didactics kit for integrating the computer into education) with which we are experimenting among primary school teachers.

This material is derived from our concepts for the teaching of mathematics at the primary level and the use of the computer. These concepts are based on our experience both in research (into mathematics learning and procedural thought learning, with the help of an FCAR grant) and in the teaching, training and development of mathematics teachers (teacher training course at the University). The educational process and the mathematics content proposed relate to the orientations of the mathematics program of the Department, in which particular attention is given to: the importance of geometry, the importance of problem-solving, the constructivist approach to learning, demystification of the role of errors in the development of knowledge, the importance of links between the various mathematics disciplines.

This material takes the form of a sequence involving the exploration and construction of mathematical concepts; these are activities in which the child works at times on the computer and at times on paper (using a pencil, ruler, protractor and compass). The emphasis is on individual or team activities, explanations by the teacher or group discussions.

The proposed paper will give a detailed presentation of this project and an initial analysis of the reactions of teachers and pupils.

COMMENT DES ENSEIGNANTS DU PRIMAIRE S'APPROPRIENT UN LOGICIEL ÉDUCATIF

H. Kayler
Maîtrise en enseignement au primaire
Université du Québec à Montréal
Montréal, PQ

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

HOMEWORK ASSIGNMENTS VIA THE COMPUTER

D.E. Stenton
Dept. of Physics
University of Victoria
Victoria, BC

In the traditional approach, assignments and tests have been used as the most significant measure of whether or not a student has achieved a certain mastery level. Assessing a student's progress is done periodically, since the process of grading each student's performance is tedious and time consuming. The feedback of results can be frustrating from the student's point of view if this process takes weeks to process. An alternative approach to this procedure is to use the Problem Delivery System.

The Problem Delivery System (PDS) provides an instructor with an alternative approach to assigning problems to individual students and having their answers checked by a computer. PDS uses a computer to randomly select a number of problems from a problem data base, assign numeric data to each problem and send the entire set of problems to a printer. A student, upon returning to enter answers, will have each problem marked and graded, and the results displayed and stored as a permanent record of their achievements. PDS contains a number of 'programming' tools to be used to create a problem data base, to monitor students' progress and to deliver and mark students' assignments. The instructor will be required to use these tools to create either qualitative or quantitative types of questions within a problem data base.

Examples will be given to illustrate the process of issuing computer managed weekly Physics problem assignments.

ATTRIBUTION DES EXERCICES PAR ORDINATEUR

Dans l'enseignement traditionnel, les exercices et les examens constituent les principaux outils de mesure du degré de compréhension d'un étudiant. L'évaluation des progrès des étudiants est un exercice périodique, car la cotation du rendement de chaque étudiant est une tâche harassante et laborieuse. En outre, les étudiants sont frustés puisque souvent ils ne connaissent leurs résultats que quelques semaines plus tard. Une solution de remplacement à cette démarche traditionnelle est le système d'attribution des problèmes.

Le système d'attribution des problèmes (PDS pour Problem Delivery System) se veut une autre façon d'attribuer des problèmes à un étudiant, dans laquelle les réponses sont corrigées par ordinateur. Le PDS fait appel à un ordinateur qui choisit entièrement au hasard un certain nombre de problèmes dans une base de problèmes, décide des valeurs numériques et achemine l'ensemble des problèmes à une imprimante. Les réponses de l'étudiant sont corrigées et cotées dès qu'elles sont entrées, et les résultats sont consignés et conservés dans un dossier d'évaluation permanent. Le PDS renferme un certain nombre d'outils de programmation pouvant servir à constituer la base de problèmes, à surveiller les progrès des étudiants et à envoyer et coter les exercices. L'instructeur a besoin de ces outils pour ajouter dans la base de problèmes des questions à développement ou à calcul.

Nous donnons ici des exemples du processus informatisé d'attribution hebdomadaire de problèmes de physique.

HOMEWORK ASSIGNMENTS via the COMPUTER

D. E. Stenton

Dept. of Physics and Astronomy,
University of Victoria, Victoria, B.C.

SUMMARY

The Problem Delivery System offers an alternative approach to issuing problem assignments on a regular basis to a class of students. Those features which provide a significant benefit to both the instructor and student are summarized as follows:

1. **GENERIC SYSTEM** - since problem data and program routines are independant, any course or subject material can be installed as a problem data base.

2. **FLEXIBLE PROBLEM TYPES** - questions within each problem can be defined as either qualitative or quantitative.

3. **COST EFFECTIVE** - one computer terminal can serve as many as 50 students.

4. **UNIQUE ASSIGNMENTS** - all students receive a random selection of problems from the problem data base; each question contains a unique set of numerical data.

5. **EFFICIENT OPERATION** - problem delivery, problem marking and assessment are immediate.

6. **PORTABLE OPERATION** - all problem data, students registration and library files are totally portable between the operating systems, DOS and UNIX.

7. **STUDENT FEEDBACK** - all students receive question answers and a marks summary after completing each assignment.

8. **INSTRUCTOR FEEDBACK** - all instructors have access to the students performance records to examine levels of difficulty, to compare class performance, to determine access times, etc.

9. **PROBLEM PRECISION** - format used to design the problem data base tends to limit the assigning of ambiguous questions.

10. **INTERDEPENDANT** - the system is designed not to be operated independant of lectures, discussions, laboratories, seminars, tutorials, etc.

PROBLEM DELIVERY -- WHAT IS IT?

The Problem Delivery System (PDS) is a convenient support utility for any course which requires that a problem assignment be issued to an individual student and a record be made of a students performance in solving a problem. Assignments are generated from a data base of problems which reside in a computer. The computer may either be a personal computer, such as an IBM-PC, or a mainframe computer such as a VAX 780, as long as the operating system is either DOS or UNIX.

Problem assignments are issued via a printer and are randomly selected, usually from a large problem set. Each set is distinguished from all others by the allocation of a week assignment number which precedes the name of the problem data. A student may choose a number of problems he wishes to receive (up to a maximum of 5) at any given time, but he must return to the computer to answer these problems before requesting another assignment.

PROBLEM DELIVERY -- HOW DOES IT WORK?

The procedure for receiving an assignment from PDS is very simple since all instructions are issued as menu commands. Once the student has 'signed-in' with their name and account number, they will receive a prompt to enter the number of questions they wish to receive at this time. The maximum allowable is set to 5. This is the only decision the student is permitted to make since PDS then begins to sort through its problem data base in preparation to send to the printer the requested number of problems.

Just prior to the time of printing, the computer generates some random numeric data within each quantitative type question. It then assigns each data to specific locations within the problem, or question(s), in preparation for delivery. This

data, called the question data, is stored in a file under the students identification number, along with the answer data, question number and week assignment number. Each student therefore receives a unique problem assignment and a record is made of their specific data which has been assigned to their problems.

At the conclusion of the printed copy, the student is automatically logged off the system. This procedure requires that a student would normally spend a minimum amount of time at a terminal, or microcomputer, particularly during delivery. The computer is therefore available for the next student to log-on to the system.

If a student 'logs-on' to the computer to begin answering an assignment, he will first receive a prompt asking for a reply to each question. Each time the student answers a question, it is compared to a previously stored answer. If the answer is not the same, the computer begins to test and compare possible errors which are related to the solution. This may be an answer unit in the case of a quantitative question or a test for spelling errors in the case of a qualitative question. The precise answer to each question is acknowledged, and both the student's answer and correct answer are sent to the screen and printer. This information is stored under another specific file name, using, in part, the student's identification number.

If the student replies with an incorrect answer, a list of five students' names are displayed on the screen. These students are selected from a HELP file (a file containing students who have correctly answered the specific question). The names of these students are displayed for the purpose of consultation as to 'how to do the problem correctly'. The HELP file represents a form of peer learning.

It is possible for a student who wishes to quit the system before completing the whole assignment to return to answer any of the incomplete problems without any penalties. There is, however, a limit of three attempts to solve a problem.

AUTHORING TOOLS

Although the problem delivery system is essentially menu 'driven' and requires very little programming experience, it does require some understanding of the DOS or UNIX operating system in order to arrange the problem files in a manner suitable for delivery.

Initial entry into the system requests the user to enter the name of the course instructor, name of the course and course number. Essentially, this initial procedure provides a small amount of security which may deter non-authorized users from accessing problem data or students' accounts.

Once an instructor has entered the system, a choice is offered to either to build a problem data base or to run the delivery system. Selecting the Build procedure will prompt a second menu to appear, permitting the author to create a problem data base, change the SI units library, create a special library, register or display student marks, or simply display any data file. Each program within the Build procedure contains easy-to-use prompts. For example, information on students registered in a course can be sent to the screen, or to a printer, instantly. A summary of all students accumulated marks and potential marks for all assigned problems is therefore available at any time.

INTELLIGENT REPLIES

The duties performed by PDS, in most cases, simulate the approach taken by a human tutor. PDS is capable of recognizing incorrect answers on several levels of non-acceptance and inform the student of accepted responses. When a student replies to questions in any given assignment, the answer will be considered incorrect if units are not included. An answer to a problem which contains a numeric value must include a unit, specifically, a properly abbreviated unit, if it is to be truly meaningfull. If a unit is not properly abbreviat-

ed, i.e., m for metre, but is correctly spelled, a 10% deduction to the mark is given. If this is not acceptable, a spelling check will be made [meter, metres, etc] and another deduction made to the mark. If this is not satisfactory, a search begins for a unit conversion, which would be (in this case) cm for centimetre, A for angstroms, AU for astronomical unit, etc, and the proper numeric conversion (i.e.,100cm) is computed. Assuming the units have been correctly identified, the students answer is compared with the correct answer and a mark awarded if the numerical portion lies within 2% of the correct answer.

Using the prefix abbreviations (Mega, kilo, milli, etc) in front of the unit will not confuse the system. The program 'strips' this information then converts the numeric value with the appropriate power (10^6, 10^3, or 10^{-3}). Of course, one must use engineering format (i.e., 2.56E-34) when entering a large (or small) number into the computer terminal.

Part marks are awarded if the program determines that the method of solution appears correct, but the answer is numerically wrong. An answer to a question which requires the use of a previous answer is automatically recognized by the program, and a computation is made using the students data. A student would therefore receive part marks for the correct method. The procedure to determine if the unit is correct is the same as described before.

HARDWARE CHOICES

NATAL, the authoring language used to develop problem delivery programs, operates within the UNIX environment and the DOS operating system. This makes the problem delivery system very flexible in many respects. It can be accessed via a terminal and main-frame computer, providing a totally secure system. It can be used in a master-slave configuration with an IBM PC, XT or AT (or IBM clones) network of computers or workstation environment.

HOMEWORK ASSIGNMENTS

There are advantages and disadvantages to be considered in making a choice between the delivery of problems by microcomputer versus a mainframe computer. Interestingly, a combination of the two systems, a 'home' microcomputer and a central mainframe computer system, offers the best of both worlds. Using an inexpensive modem, homework assignments can be obtained from a central networked computer system and information can be transmitted back to this system efficiently, conveniently and, in most cases, with no end-user cost. Essentially, the mainframe system provides security and maintainance that is generally not available or not practical to implement on a microcomputer.

THE POTENTIAL FOR HAVING LARGE "POP-UP" EDUCATIONAL MENU SYSTEMS

W.D. Hayhoe
Dept. of Measurement, Evaluation and Computer Applications
Ontario Institute for Studies in Education
Toronto, ON

Menu systems are particularly helpful for infrequent users because they structure the choice space and require little memorization. Menu panels usually contain only a few items, however, and this has limited the use of menu systems to databases with under 100 items. Empirical and theoretical work now shows that menu panels of up to 100 items (and menu systems of several thousand items) are feasible, if the items are properly categorized. Experimental work at O.I.S.E. studied the categorization of menu items in large panels of 48. "Educational" datbases consisted of clothes, furniture, occupations and sports. Three methods of categorization were compared: single expert, each subject for her own use, and analysis of sorting data from 48 typical subjects. Expert categories gave more miscategorization errors than sorting-based categories. Mathematical analyses monitored the relative effect of the different structural variables. The implications for education are as follows: A large pop-up menu system of several thousand world geography items or French grammar items appears quite feasible, if the items are properly categorized in large menu panels. Average retrieval time per item might be as low as 15 seconds. Also, the studies indicate that developers of educational software can profit from a study of the "perceived world view" of the users by a sorting analysis.

POSSIBILITÉ D'AVOIR DES SYSTÈMES PILOTÉS PAR DES MENUS VOLUMINEUX APPELABLES DIRECTEMENT À L'ÉCRAN DANS LE DOMAINE DE L'ENSEIGNEMENT

Les systèmes pilotés par menus sont particulièrement utiles aux utilisateurs occasionnels parce qu'ils structurent l'ensemble des options et n'exigent que peu de mémorisation. Les menus, cependant, ne contiennent habituellement que quelques options et cela a limité l'utilisation des systèmes pilotés par menus aux bases de données ayant moins de 100 éléments. Des travaux empiriques et théoriques démontrent maintenant que les menus de plus de 100 options (et les systèmes pilotés par menus de plusieurs milliers d'options) sont possibles, si les options sont bien catégorisées. Dans un travail expérimental à l'O.I.S.E., on a étudié la catégorisation des options dans des menus volumineux comprenant 48 options. Les bases de données "scolaires" comprenaient des vêtements, des meubles, des occupations et des sports. On a comparé trois méthodes de catégorisation: expert unique, chaque sujet pour sa propre utilisation et l'analyse qui consiste à classer des données provenant de 48 sujets types. Les catégories d'experts ont occasionné plus d'erreurs de catégorisation que les catégories fondées sur le classement. Les analyses mathématiques ont contrôlé l'effet relatif des différentes variables structurales. Les répercussions dans l'enseignement sont les suivantes: un système piloté par des menus volumineux appelables directement à l'écran et comprenant plusieurs milliers d'options relatives à la géographie ou à la grammaire française semble très faisable si les options sont bien catégorisées dans des menus volumineux. Le temps d'extraction moyen par option pourrait être aussi peu élevé que 15 secondes. Les études indiquent également que les réalisateurs de didacticiels peuvent bénéficier d'une étude du point de vue des utilisateurs grâce à une analyse de classement.

THE POTENTIAL FOR HAVING LARGE "POP-UP" EDUCATIONAL MENU SYSTEMS

W. D. Hayhoe

Ontario Institute for Studies in Education
Toronto, Ontario

1. Introduction

With modern hardware, "pop-up" menus have the potential of accessing large databases, if some of their present drawbacks can be overcome. The implications of this for education are interesting. In this paper, I review empirical results of menu research, describe a series of recent experiments conducted at the Ontario Institute for Studies in Education, and discuss the implications of these findings for large "pop-up" educational menu systems.

1.1 Review of Menu Factors

In my thesis [2], I identified over 40 menu design variables, and gave the results of over 50 menu experiments carried out over the last 10 years. In this paper I will comment on some 20 of these that may be the most significant for menus in educational software.

Fig. 1 displays these 20 different menu factors organized as database, task, menu, computer, and presentation variables. Following is a brief list of empirical results about the interaction of these factors with each other:

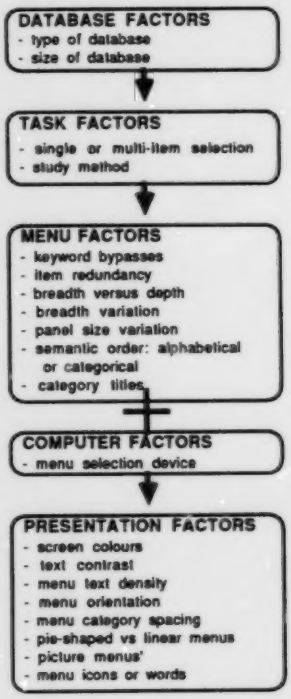


Fig. 1 Menu factors

- type of database: it is harder to form menu categories for functions than for nouns. Alphabetical order may be better in this case.

- size of database: the size increases the search time but may not interact with other menu factors.

- multiple-item menu selection: this accentuates the effect of other menu factors such as categorization.

- study method: study of a global tree diagram is better than the study of menu frames, selection sequences, or trial and error practice selection.

- keyword menu bypasses: these may facilitate menu selection especially for databases with multi-word items.

- item redundancy: some form of this in the menu system may give faster selection times.

- breadth versus depth: for short menu items grouped in categories, menu breadth (i.e., items per panel) for fastest selection may be as high as 100. For long multi-word menu items, optimal menu breadth may be 4 to 8.

- breadth variation: if the item scan time is higher at one menu level, the menu breadth should be smaller.

- panel size variation: variation in the number of menu items per panel at any one level likely has little affect on average selection times.

- semantic order: if users have a definition rather than an explicit target in mind, categorical order may be faster than alphabetical.

- category titles: titles made with the input of typical users may be superior to those designed by professionals or experts.

- screen colours: colours used to delineate menu categories may not affect selection time or accuracy.

- text contrast: a high contrast between text and background appears to improve performance.

- menu text: a suitable mix of upper- and lower-case letters is best.

- menu orientation: vertical menus are faster than horizontal menus.

- menu category spacing: screen spacing between menu categories increases selection times.

- pie-shaped menus: these may give shorter selection times than linear menus, but use more screen space.

- picture menus: adding pictures gives fewer errors but longer scan times, giving the same selection times.

- menu icons: using abstract icons rather than text may speed up menu selection if the icons are well known, but may not be remembered.

- menu selection device: the mouse is 15% faster than the trackball, and both are superior than the joystick. For experienced users, keyboard selection may be superior to mouse or trackball. Touchscreen may also be superior to pointing devices, but has a lower resolution. Eye-trackers are about 30% faster than the mouse but are very expensive, and have lower screen resolution.

1.2 Large "pop-up" menu systems

With the new generation of computers with screen resolutions of a million pixels (i.e., 1150 by 900), it became feasable to think in terms of "pop-up" menu panels of 100 or more items. A menu of several levels could then house a database of several thousand items. However, the key problem of categorization still remains. The experiments reported in the following sections concern these two areas: the structural question of breath versus depth of menu system, and the semantic question of proper categorization.

2. Menu experiment with 144 countries

A Macintosh-like "pop-up" menu structure was designed for use on the Unysis ICON, with selection made by trackball. The previous panels remained in sight, and the previous choices in the retrieval path remained highlighted. In this first experiment, countries were divided up into categories by the investigator. Three menu systems were formed of 144 countries - 12x12 (Fig.2), 4x6x6, and 4x3x3x4 (Fig.3) - and three menu systems of 144 integers in word form.

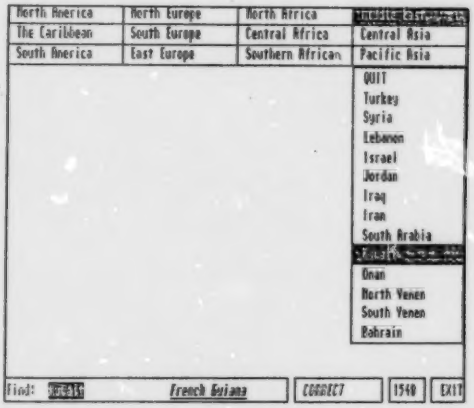


Fig. 2 Two-level countries menu

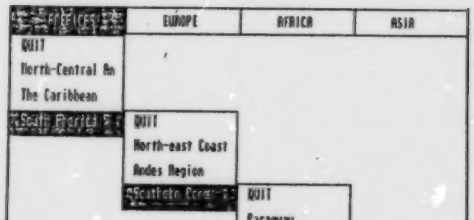


Fig. 3 Four-level countries menu

Each of 16 subjects (University of Toronto graduates) made 144 selections, 24 from each of the six conditions. For countries, selection times increased linearly with depth, from 14 s (two-level) to 26 s (four-level). Likewise errors increased linearly from 1 errors/selection (two-level) to 4 errors/selection (four-level). For the 144 integers, there were almost no errors, as expected, and selection times increased linearly from 6 s to 9 s.

3. Menu experiment with categories

The previous experiment demonstrated the potential for large "pop-up" menu systems. If 144 (12x12) integers can be accessed in an average of 6 seconds, then 1728 (12x12x12) integers can be accessed in 9 seconds, and 20,736 (12x12x12x12) integers can be accessed in 12 seconds. However, it also demonstrated the importance of proper categorization, for if integers took 6 seconds to be selected on the average in the two-level system, countries took 14 seconds. The difference increased further for the four-level system. The subjects commented freely on the difficult categorization of the countries, especially in the four-level system (Fig. 3). This difficulty may have been particular to the geographical nature of the database. Therefore, the second experiment, which studied further the depth versus breadth issue, together with the way of forming menu categories, used four different databases [2].

3.1 The categorization of the databases

Four common "educational" databases were chosen, clothes, furniture, occupations, and sports (Battig and Montague, 1969), each containing 48 database items. Together they formed a database of 196 items. They were formed into menu categories in three different ways:

1) each user forms his/her own categories
2) a software expert forms the categories
3) a group of representative users sort the database, and the categories are formed from this sorting data.

It was hypothesized that the sorting-based categories, representing typical user perceptions of the databases, would be superior to the others.

To form the sorting-based categories, 48 university students were asked to sort the 48 items in each database into "logically-related" categories and label the categories. This took 1 hour. Each subject worked with the four databases in a different order. The results were then summed over the 48 subjects, similarity matrices formed, and categories formed using hierarchical clustering (Ward's method), and latent partition analysis.

Category titles were applied to these sorting-based categories by the investigator (using the input of the 48 sorters), and also by each subject in the following menu experiment. This gave four categorization conditions altogether.

3.2 The menu experiment

In the actual menu experiment, the 16 different database by categorization combinations were placed in two different menu systems: one-level (Fig. 4) and two-level (Fig. 5). The list of of menu results given in section 1 were applied to the menu systems: i.e., spacing, grouping in categories, etc.

Another group of 24 university students was hired to complete three practice sessions. Each subject was exposed to 4 categorization/database by 2 menu depth conditions, in a counter-balanced manner. Each subject made 1 selection for each of the 384 cells, in each practice. A week before the subjects did the experiment, they studied, titled and studied, or categorized each database.

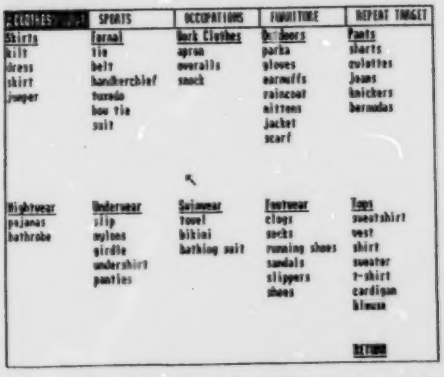


Fig. 4 One-level menu for clothes

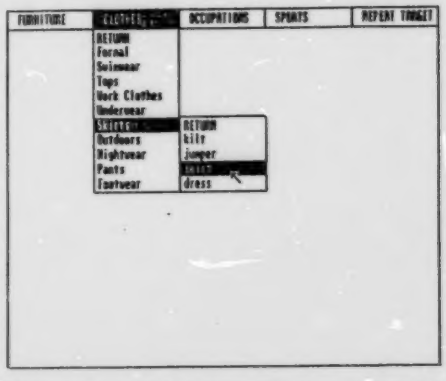


Fig. 5 Two-level menu for clothes

Selection times were shorter for the broad one-level menu (5.5 s), than for the two-level menu (6.3 s), although these differences decreased with practice. The sorting-based categories with investigator titles were superior to to the other categorizations, in selection times, errors, "goodness of fit" ratings, and memory recall errors.

3.3 The menu experiment repeated

In the previous experiment, the 24 subjects all studied or sorted the four databases one week earlier. This was necessary in order to compare the self-sort/self-title categorization condition with the others. It was found that this condition was not as efficient as the user-sort categorization.

However, the results obtained by comparing the sorting-based categories with the expert categories may have been affected by the previous differential exposure of the subjects to the categorizations. Therefore the experiment was repeated in order to compare the sorting-based (investigator titles) with the expert categories. Ten more students were hired, but were not exposed to the categorization/database conditions before the experiment. The error rate in the two-level menu (0.8 errors/selection) was higher than in the previous experiment (0.5 errors/selection). The selection times were also higher than previously: 8.4 s versus 6.5 s (two-level), and 6.4 s versus 5.5 s (one-level). In the second practice session, the errors decreased to 0.25 and the selection times to 5.0 s. It was demonstrated that the sorting-based categories were superior to the expert categories in selection times, errors, and recall errors at a later date.

Fig. 6 displays the selection times versus error rates for the two-level menu system, throughout the 32 blocks of 12 trials each, summed up over ten subjects. It can be clearly seen how much the selection times are dependent on the number of errors made.

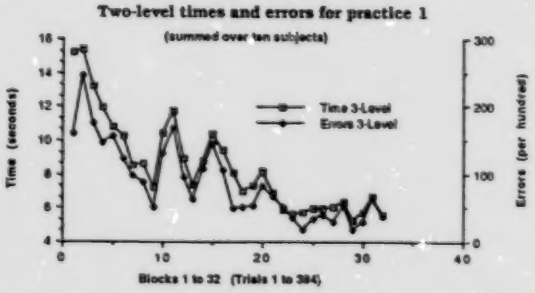


Fig. 6 Times and errors compared

A detailed analysis of errors was conducted, and it was demonstrated that the expert categories contained more "miscategorization" errors and "vague category labels" than the sorting-based categories. However, both categorization systems contained overlapping categories.

4. Discussion

These experiments, conducted with "pop-up" menu systems using databases of 144 and 48 (or 192) items, gave average selection times of 6 to 8 seconds when the categories were known. The broadest menu panels gave the lowest error rates and the fastest selection times. In the first experiment the large panel breadth was 12 items and in the second 48 items.

The principal problem in using menu systems of several depths was that of categorization errors. This has a great effect on selection times, increasing them by 300% in some cases. However, it was demonstrated that sorting-based categories, based on the perceptions of a typical group of users, can reduce the categorization errors down to a half. The remaining categorization errors are due mostly to overlapping categories inherent in the database. I suggest that a mild form of "item redundancy" where about 10% of the menu items are placed in two or more locations in the menu system might further reduce these errors by another 50%. We are now testing this.

If the problem of categorization can be solved, or at least reduced to a small fraction of what it presently is, then a "pop-up" menu system of several levels of 45 items each may hold thousands of items. For example, a 45x45 menu system hidden under one panel at the top of the screen could allow 2000 menu items to be accessed in about 10 seconds. These items might be 2000 world geography items or 2000 sample physics problems.

The sorting-based categories could be formed by having several different classes of typical students sort the database items into groups and then apply a cluster analysis. Of course, 2000 items will take longer to sort and classify than 192 items -about 10 times longer - but it is not impossible. The sorting could be divided up among several classes, and methods devised to reduce the work to several hours. It would also be an educational learning experience for the sorting students.

References

1. Hayhoe, D. The selection of database items from hierarchical "pop-up" menu systems. The Ontario Institute for Studies in Education, 1986.

2. Hayhoe, W. D. S. The development and evaluation of sorting-based categories in computer command menus. Unpublished doctoral dissertation. University of Toronto, 1988.

QUALITY ASSURANCE FOR COMPUTER-BASED INSTRUCTION

L.C. Christensen, W. Hays
Computer Science Department
Brigham Young University
Provo, UT, USA

M.R. Bodey
AT&T
Denver, CO, USA

Since early 1984, the Computer Science Department at Brigham Young University has been involved in the automation of beginning programming classes and varied projects in software and courseware development. One successful project, The Elrond Project, is a computer-aided instruction system whose purpose is to teach students about programming and computers in general, without human instruction. In the projects we have noticed many inherent differences between courseware and software. The subtle differences go unnoticed or, at least, unaddressed in many courseware projects.

This paper presents our techniques in creating quality courseware. We have built our courseware program around techniques used in software quality assurance and present some ideas that will help courseware developers create a quality product.

ASSURANCE DE LA QUALITÉ DE L'ENSEIGNEMENT AUTOMATISÉ

Depuis le début de l'année 1984, le Computer Science Department de l'Université Brigham Young oeuvre dans le domaine de l'automatisation des cours de programmation pour débutants et dans divers projets d'élaboration de logiciels et de didacticiels. L'un des projets fructueux, le projet Elrond, est un système d'enseignement assisté par ordinateur en matière de programmation et d'ordinateur en général sans intervention humaine. Dans les projets, nous avons remarqué de nombreuses différences inhérentes entre le didacticiel et le logiciel. Les différences subtiles passent inaperçues ou, à tout le moins, ne sont pas prises en compte dans de nombreux projets de didacticiels.

Dans le présent document, nous exposons les techniques que nous utilisons pour créer des didacticiels de qualité. Pour ce faire, nous avons eu recours aux techniques utilisées dans l'assurance de la qualité des logiciels et nous allons exposer certaines idées qui aideront les concepteurs de didacticiels à créer un produit de qualité.

QUALITY ASSURANCE FOR COMPUTER-BASED INSTRUCTION

Larry C. Christensen
Bill Hays

Computer Science Department
Brigham Young University
Provo, Utah

Michael R. Bodey

AT&T
Denver, Colorado

The majority of courseware products available today are not designed nor matched to instructional performance objectives. This paper addresses the issue of software quality assurance and presents a plan by which developers can create quality courseware. This paper will assist researchers to gain insight into the research-to-product phase of development and insure quality courseware.

INTRODUCTION

Since early 1984, the Computer Science Department at Brigham Young University has been using computer-aided instruction with beginning programming classes and varied projects in software and courseware development. The Elrond Project[2] is a computer-aided instruction system whose purpose is to teach students about programming and computers in general without human instruction. In all of the projects we have noticed many inherent differences between courseware and software. The subtle differences go unnoticed or at least unaddressed in many courseware projects.

Many successful software projects use five distinct organizations: Requirements, Documentation, Development, Test, and Quality Assurance. We recognize the value of these units but when developing courseware, we find a different life cycle emerging. Our project phases include:

High level design (DR1)
Low Level design (DR2)
Code phase (C1)
Test Phase (C2)
Monitor and Maintenance Phase

The following explains each of these phases of courseware development.

REQUIREMENTS PHASE

A document called the Statement of Requirements describes the courseware goals, concepts, and objectives. The developer writes this statement. This phase should be completed for all lessons in the course before any software development takes place. The Requirement Phase is a major subset of the basic requirements document.

The Requirements Phase translates the objectives into external specifications. This represents the system's behavior from the user's point of view. The nature of the product has now been clarified and product goals identified. The next step is the translation of the external specifications into High Level Design (DR1). These specifications serve as a basis for the products created by the Development organization.

HIGH LEVEL DESIGN (DR1)

Given the courseware goals and lesson requirements, the developers now define a high level user interface which includes the following:

1. Define the sequence of screens for a given lesson.

2. Identify each panel uniquely. This requires a naming convention for each panel. L1P5 would mean Lesson 1 Panel 5, or the 5th panel displayed in Lesson 1. This will allow easy tracking of panels later in the testing cycle.

3. Developers design and plan for testing. This means creating a testing sequence which executes when the lesson has been completed. The testing sequence is a simple path test at the beginning. This list grows as the life cycle progresses and becomes an on-going process.

At the conclusion of the High Level Design phase, the document is brought before a design review committee. A moderator conducts and a scribe records all notes, recommendations, action items, and modifications. The document is then given back to the original team that will modify the design. When the document is modified, it is redistributed with the original copy to the DR1 committee members and they sign it off. The developers then begin the DR2 phase.

LOW LEVEL DESIGN PROTOTYPE PHASE (DR2)

Each phase offers increasing levels of detail. The first phase, Preliminary Functional Design, translates the High Level Design into functional specifications. The text, notes, graphs, and story-boards relate to each panel. These panels are linked together. This provides a prototype shell of the given lesson using a standard template panel for orientation purposes. Menu structures, the use of graphics, the design of screens for clarity, and other design strategies are resolved at this stage[3].

The next step is to verify that the prototype shell is working and all valid function keys are performing their specified task defined in the project's Requirements Phase.

The Committee conducts another review and use part of this review as a demonstration via the computer with the prototype shell. During the review, someone other than the originator runs the program. They record and correct any modifications before signing off.

With this prototype shell, the programmer knows that the desired sequence takes place in the lesson. If problems in the lesson develop later, the errors are found in the code and not in the design of the lesson.

CODE PHASE (C1)

Given the DR2 document, the programmer is ready to create the software for the given lesson. Previously defined are the data structures, sequence of screens, type of interaction, logic flow, and the prototype lesson shell. The programmer now generates the executable code for the course of study. We attempt to impose standards in this phase but find many of our programmers desire to insert their own creativity. We impose strict standards in this phase to discourage modification standards.

The C1 Phase encompasses coding, debugging, and preliminary testing of each lesson. Testing is easier when a scheme for coding has been previously defined that includes indentation and comment standards. Corrections are made by the programmer and all committee members sign off the C1 for the lesson. Upon completion and review of the C1 Phase the lesson is ready for C2.

TEST PHASE (C2)

This phase is composed of four different sub-phases each with specialized objectives: Integration, Functional, System, and Regressions. An important strategy is to modularize the effort by dividing the testing effort into manageable components. The relationship between phases and modules is shown in Figure 1.

Integration:	White-Box Module Crash Module
Function:	Functional Test Module
System:	Human Factors Module Supportive Materials Module Pedagogy Module Field Module
Regression:	Regression module

Figure 1
Module Composition of Test Subphases

Testing courseware is easier in small units, usually a lesson at a time. Testing on this level emphasizes the verification of logic, algorithms, data handling, student pacing, functions of the lesson and its performance. We utilize automated white-box techniques to examine the internal structure of the program. Black-box techniques complement the automated tools in providing full coverage. Research done by the authors indicates that over one third of the errors found during the Test Phase compare favorably to the 35% unit test figure given by Boris Beizer and others[1]. Errors generated here require a Program Trouble Memorandum (PTM) and are sent back to the Implementation phase (C1) for correction. PTM's specify the nature of the error, describe the keystroke path, the hardware environment, date and time, screen description, version ID, and name of tester. Figure 2 is an example of a Program Trouble Memorandum.

PTM#_____
Product Name:_____
Version #:_____
Tester Name:_____
Date Found:_____
Hardware Configuration:_____
Keystroke Path:

Frame Description:

Error Category:_____
Problem Description:

Priority ID	Description
1	Fix necessary to continue with testing error causes program crash
2	Fix required but not immediately necessary fix by next version
3	Fix required by end of testing phase moderate error
4	Error acknowledged but no authorization given to fix
NR	Nonreproducible error

Figure 2
Program Trouble Memorandum & Error Priority

The second test phase begins as soon as a sufficient number of lessons from the course have been developed. Lessons are packaged together and then tested to find discrepancies between the intent of the external specifications and current version of the courseware. This test verifies that the initial objectives of the product have not been compromised. The comparison of the original Statement of Requirements and the specifications achieves this test.

The final phase is the Regression Test. Regression testing is a comprehensive retest of the failed areas in the product and is performed on final versions of the product.

MONITOR AND MAINTENANCE PHASE

Depending upon the life span of the product, the Monitor and Maintenance Phase may account for 50% of the life cycle costs[1]. This phase continues throughout the life of the product and also future versions of the product. This phase is responsible for the recommendations for a new version - the new features to be considered and what old features need to be revised or deleted.

A comprehensive documentation of the history of decisions made during the development and testing of the original product is next. Any modifications made during the final phases, require the product to undergo new requirements, development, and testing. Usually, only those portions of the system affected by the modification need to be retested. However, changes at a given level will necessitate retesting and reverifying products, and updating documentation at each level.

This Phase collects and reviews end-user requests and complaints, bugs that escape detection, and current usage statistics.

COURSEWARE TEST GUIDELINES

The quality assurance activities is divided into quality management and preparations for reviews and software verification and validation[4].

Software verification and validation include: walk throughs or code inspections, white-box and black-box testing techniques, and unstructured (crash or ad hoc) testing procedures. These testing methodologies surface many different types of errors. Error classification studies provide an understanding of the different types of errors and provide insight into probable detection methods[6].

Similar types of errors are found in Boris Beizer's book: Software Testing Techniques, and Glenford Myers' book: The Art of Software Testing[5]. All three lists are comparable. The lists differ in terminology and category partitioning.

It is obvious that courseware error classes should embrace the eight classes found in Yourdon's and also include the following:

1. Human factors errors: Present data in a concise manner. Appendix A contains a list of human factor issues that should be considered when creating courseware.

2. Pedagogical errors: Determine the lessons' educational value and that the displayed data follows the instructional design specifications. We present pedagogical issues in Appendix B. This list is useful for designing test cases for pedagogy and instructional design.

3. Data Errors: The data volume is massive in courseware. A typical diskette courseware product contains 500 to 1000 frames of information and the data is present in the basic product. In contrast, software contains little intrinsic data.

4. Navigation errors: Since the end-users traverse the product frame by frame, navigation specifications present a prime location for errors.

CONCLUSIONS

Present software testing techniques have proven to be inadequate in testing courseware and must be modified to accommodate courseware development. The modular testing framework and sign off criteria for courseware development utilizes relevant software techniques. Human Factors testing must focus on capturing understandability and usability errors. Customize pedagogy testing for courseware so attention is focuses on instructional design, pacing, presentation, and other pedagogical issues. The System Test examines the data and paths associated with many frames peculiar to courseware. The Field Test demonstrates the value and correctness of original premises.

Quality assurance in courseware development requires considerable creativity, diligence, and patience. Testers must avoid a "production line" mentality. Psychological factors are as equally important as technical factors in designing courseware and especially in designing and executing the test plan.

APPENDIX A - HUMAN FACTORS ISSUES

I. MEMORY AND ATTENTION
- Performance
- Amount of text on screen
- Complexity of Graphics
- Screen formatting criteria
- Placement of information on screen
- Presentation style (Concepts, Question, and Instructions presented one at a time)
- Timing (user Control)
- Computer response time constraints
- Error Messages (statement of the problem and possible solutions)
- Supportive materials (Maps, Tables, Figures, Advanced Organizer, Reference card, etc.)
- Acceptable Attention-Span constraints of average age of users
- Questions that produce lone-term memory
- Prompt active keys
- Ability to progress and regress within product
- Consistent Format
- Help available (Function oriented)
- Hint available (Content oriented)
- Closure of ideas and concepts
- Navigation information available

II. LANGUAGE / TEXT CHARACTERISTICS
- Proper level vocabulary
- Sentence structure, sentence length, etc.
- Clause structuring of paragraphs vs. blocked text.
- Size of text
- Type style or font
- Proper names, places, things (Defamation testing)
- Preliminary discussion of all symbols, terminology
- Glossary available
- Use of jargon, acronyms, mnemonics, and abbreviations, contractions, stacking, etc.
- Use of humor
- Use of first person and student's name consistent presentation
- Use of concrete concepts vs. abstract concepts
- Multiple presentation (sound, sight, etc.)
- Enlarged text, Voice, graphics
- Use of inquiry techniques
- Reinforcement of concepts
- Use of dialogue and questions

III. GRAPHICS AND VISUAL PROCESSING
- Use of color (to code, differentiate, highlight, make real)

Palette
Color (foreground / background)
Picture resolution
Cursor size, positioning, shape
Use of bit-mapping vs. draw-and-fill techniques
Use of cuing (arrows, color, highlighting, blinking, flashing, underlining, etc.)
Graphics clarity
Graphics standards (ethics / morality)
Graphic descriptions or analogies

IV. CONTROL CHARACTERISTICS
Menu philosophy
Menu screen design
Pop-up window philosophy
Option features (Sound, color, voice, etc.)
Function features (Exit, escape, return, back, forward)
Auxiliary features (Printer, screen print, etc.)
Expert / Novice strategies
Hierarchical control strategy
Pacing
Check point restart

V. INPUT / FEEDBACK
Interaction standards
Feedback standards
Numerical input standards
Alphanumeric input standards
Input prompts and fields
Clearing erroneous input methods
Erase standards
Error recovery (Data entry, system failure, conceptual)
Protection for destructive functions (Exit, undo, escape, clear, etc.)
Sound / visual (see above)
Responses to right and wrong answers
Program response to students (scoring method, progress, opportunity to correct answers, etc.)

APPENDIX B - PEDAGOGY AND INSTRUCTIONAL DESIGN ISSUES

Target audience specified and appropriate
Prerequisites specified
Objectives accurately stated for course, units, lessons, activities
Objectives appropriate for grade level
Pre-instructional strategies used
Scope and sequence accurately stated
Input style appropriate
Internal design and function consistent
Graphics appropriate and relevant
Cuing and prompting used consistently and properly
Graphs, charts, or other visuals included in the program--well designed and technically correct
Effective use of color
Effective use of sound
Content appropriate to abilities and interests of target audience
Appropriate response times
Helpful "HELP" screens
Interaction used effectively
Gender balance
No stereotyping
Excessive competition and violence avoided
Consistent with major basales
Program's expectations of students realistic
Congruent with target curricula
Degree of difficulty of activities
Sequencing of materials
Presentation
Demonstrations congruent with activities
Pacing or rate of display
Feedback (Non-threatening, appropriate, varied, immediate, relevant to age group)
Content motivational
Informative menu structure

BIBLIOGRAPHY

1. Beizer, Boris, Software Testing Techniques, Van Nostrand Reinhold Co., New York, 1983.

2. Christensen, Larry C., Elrond: A Computer Based Instruction System for an Introductory Programming Course, SIGCSE Bulletin, Vol.17, Number 1, March 1985.

3. Christensen, Larry C., Design Strategies For a Computer-Based Instruction System, ACM SIGCSE Bulletin, February 6-7, 1986.

4. Deutsch, Michael S., Software Verification and Validation- Realistic Project Approaches, Prentice-Hall, N.J., 1982, p.294.

5. Myers, Glenford J., The Art of Software Testing, John Wiley & Sons, New York, 1976.

6. Yourdon, Edward, Techniques of Program Structure and Design, (Englewood Cliffs, N.J.: Prentice-Hall, 1975),pp.254-256.

TEACHING MIDDLE-SCHOOL SCIENCE WITH A COMPUTER-BASED INTEGRATED LEARNING SYSTEM

T. Levin
Edunetics Ltd.
Herzlia, Israel

R. Nachmias
School of Education
Tel Aviv University
Tel Aviv, Israel

SEEDS is an integrated instruction evaluation computer-based system designed to support student learning of Junior High Science. It is designed to automate and integrate student learning needs in the context of a well defined, objective based, full 3-year curriculum. SEEDS includes instructional components which integrate different modes of learning and instruction, such as tutorial units that present facts, concepts and principles; exploration and manipulative laboratory units which focus on discovery investigation; collection of worksheets and complimentary resource catalogue.

The diagnostic assessment component of the system is a testing system designed to generate tests of different types, to score, analyze and report test results and to suggest adoptive activities for students.

In this presentation, we will present an overall view of the system and discuss the potential of integrated systems as an innovative instruction provider.

L'ENSEIGNEMENT DES SCIENCES AU NIVEAU INTERMÉDIAIRE AU MOYEN D'UN SYSTÈME INTÉGRÉ D'ENSEIGNEMENT AUTOMATISÉ

SÉÉDS est un système intégré et évolutif d'enseignement automatisé conçu pour faciliter l'apprentissage des étudiants de la Junior High Science. Il vise à répondre de façon automatique et intégrée aux besoins des étudiants en matière d'apprentissage dans le cadre d'un programme complet de 3 ans, bien établi et pour lequel des objectifs ont été fixés. SÉÉDS comporte des outils éducatifs faisant appel à trois modes d'apprentissage et d'initiation intégrés et différents tels que les unités d'initiation qui présentent des faits, des concepts et des principes; des unités d'exploration et de manipulation qui mettent l'accent sur la recherche et la découverte; un ensemble de feuilles de calcul et un catalogue des ressources complémentaires.

La composante évaluation et diagnostic du système est une fonction d'essai qui génère des tests de différents types, mesure, analyse et signale les résultats obtenus aux essais et propose des activités adaptées aux étudiants.

Dans le présent exposé, nous allons donner une vue générale du système et parler des possibilités des systèmes intégrés comme outil d'initiation innovateur.

TEACHING MIDDLE-SCHOOL SCIENCE WITH A COMPUTER-BASED INTEGRATED LEARNING SYSTEM

Tamar Levin and Rafi Nachmias

Edunetics Ltd.
School of Education
Tel-Aviv University, Tel-Aviv, Israel

INTRODUCTION

Science education is currently looking for new directions to bring about the scientific literacy of all students. One of these directions that is considered to have great potential, is the use of instructional information technologies. In this paper, we will describe an integrated learning system -- SEEDS -- that is packaged to span a large part of the science curriculum in middle and junior-high schools.

SEEDS Science Education, Evaluation and Diagnostic System is a computer-based integrated Learning System for middle and junior-high school science. It offers the advantages of using computers to diagnose, reinforce and enhance individual instruction, and to monitor students improvement.

The SEEDS lessons are highly interactive and adaptive to student learning needs, and they provide opportunities for learning science in a variety of instructional modes. The curriculum focuses on the nature and structure of science and processes of scientific inquiry. As such, it provides students with opportunities to learn basic concepts and rules of science as well as to explore, discover, and develop major skills of science inquiry.

SEEDS helps the teacher to place students along the curriculum scope and sequence, to adapt the instructional sequence and pace to each student, and to provide multimedia and multimodes of science learning. SEEDS gives diagnostic information about each student, and prescribes instructional activities (remedial, corrective or enrichment). Student performance and progress is evaluated by grading and mastery tracking.

DESCRIPTION OF SEEDS

This section describes SEEDS' structure and functions according to its two broad functional components: the instructional diagnostic component and the assessment component.

The instructional component consists of a complete junior-high science curriculum program. It integrates different modes of learning and instruction and focuses on presenting facts, concepts and principles, as well as on processes of scientific inquiry.

The structure of the learning objectives for the curriculum can be presented in two dimensions: content and cognitive processes.

The **content dimension** is presented by the division of each science area into topics and smaller elements. These elements are --

- **facts** (e.g. limestone floats on mercury; in the U.S. the days are longer than nights between 21 March and 23 September), and **concepts** (e.g. density, diffusion, symbiosis, tides).

- **sequences** or processes which refer to the order of phenomena in the sequence of their occurrence in nature or under experimental manipulation, and the causal relationships of those phenomena (e.g. the life cycle of the honey bee proceeds through successive stages from birth to death); and,

- **principles** which refer to generalizations derived and established by scientists on the basis of a large number of observations of phenomena (examples: Archimedes' principle, Mendel's Laws).

This classification is applied to each specific science area, as well as to general aspects of science pertinent to all the natural sciences. In addition, there are concept principles and processes that are related to the nature of sciences, such as hypothesis, experiment, observation, data interpretation, randomness, and inquiry procedures.

The **cognitive processes** dimension expresses the cognitive behaviors that students are expected to attain as an outcome of science instruction. Most of these categories adapted from the Bloom's "Taxonomy of Educational Objectives". Others are categories related to carrying out the processes of scientific inquiry.

The **knowledge behaviors** are typically demonstrated by recall of information presented to the student. **Comprehension** behaviors are typically demonstrated when information learned is presented in a context which differs from the context in which the original instruction was given (e.g., to translate a fact, concept or principle that is presented in one symbolic form to another symbolic form).

Application are demonstrated by students' utilization of knowledge in solving problems where the mode of solution is not specified. Application behaviors to new situations may be defined in the same field of science, in a different field of science, or to problems outside of science.

The general category of **scientific inquiry** processes focuses on the behaviors of the student involved in inquiry. It includes behaviors demonstrating observation, measurement, recognition of a problem, interpretation of data, formulation of hypotheses,

generalization, and the manipulative skills involving laboratory equipment and techniques.

The objectives of the curriculum, are defined by a combination of the content dimension and behaviors in the cognitive processes dimension

The instructional modes and means used to enhance the learning of the specified objectives are discussed below. There are several different instructional strategies, types and sources:

The **tutorial type** and the **exploration** (open) type are both on-line lessons that represent two different instructional modes intended to develop different types of skills. The **manipulative laboratory** lessons integrate off-line and on-line activities enabling students to actually perform laboratory tasks and use ready-made on-line data sheets to organize and analyze their findings. The fourth type of instructional mode is in the form of **worksheets** which suggest activities (on-line or off-line) to be used inside or outside the school. The final instructional means is the **complementary** resource catalog which includes calibrated lists of relevant available multimedia instructional materials.

The **tutorial** lesson makes use of the guided approach to learning following the concept of "direct instruction." Each such lesson covers a specific well-defined objective or small set of objectives. It presents the topic and also provides examples, demonstrations, questions and practice activities. The interactive nature of the lesson activities is manifested mainly in its branching capabilities, its demands for students' cognitive involvement and its response to students' activities.

A lesson employing the **exploration mode** defines an open environment which facilitates the understanding of the processes of scientific inquiry through exploration. These lessons utilize several models which represent the discovery approach to learning, where the students are actually simulating the scientific process. These activities provide, on the one hand, experiences with the major skills of inquiry (process based), and on the other hand, they facilitate deeper understanding of science concepts and principles (knowledge based). The integration of the basic concepts, principles and techniques is usually required to solve problems.

The **manipulative laboratory** lessons will employ both on-line and off-line activities. The lessons include:

* questions, guidelines, and suggestions for students who need to actually pefora experiments in real laboratory situations
* on-line data sheets to be filled in during experimentation
* tools or techniques for data processing

The actual experimentation or real lab situation is complementary to the simulab mode of instruction. Its intent is to strengthen the students' manual and coordination skills as they work with various tools of the scientist's trade, as well as to facilitate the students' capacity to carry out a sequence of manipulations towards a defined end.

The collection of **worksheets** will be an additional instructional means provided by SEEDS. It includes enrichment activities linked to the scientific concepts, principles and skills learned. The worksheets are used for multiple purposes, such as homework, remedial practice activities, and enrichment activities. A worksheet is designed from a collection of questions and activities which are calibrated to specific science objectives. Since it is actually a pencil-and-paper activity, the worksheet is limited in its presentation mode to printed text and graphics only. Nonetheless, it is flexible in the response mode as required. For example, worksheets are not limited to multiple choice questions.

The **resource catalog** is an instructional means which aids in adapting instruction to individual student needs for remediation, correction, or enrichment. It contains listings of relevant multi-media instruction resources e.g. film, video, experimentation kits, chapters in books, courseware. Information about each item in the catalog includes its objective, media type, mode type, recommended usage, materials needed, and the time required for completion.

THE DIAGNOSTIC ASSESSMENT COMPONENT

The diagnostic assessment component of SEEDS is a testing subsystem which contains and generates tests of different types. It is able to score, analyze and report test results and to suggest tailored learning activities for students.

The testing subsystem includes software utilities for automatic and manual generation of tests, scoring of tests, and reporting results that include scores and recommendations based on these scores. The subsystem makes use of a science item-bank and a set of test templates tailored to each test type.

The testing subsystem is general enough to be used as an independent science evaluation module. However, the specific formats of the tests it produces, the item-bank it utilizes and the data it provides will all be exploited most efficiently when linked through a management system to a rich curriculum and collection of instructional materials.

The testing subsystem offers a variety of test types aimed at serving multiple purposes. Four general categories of test types are linked to predetermined points in the instruction sequence.

The first category is **placement tests**. These tests serve as prospective assessment instruments and are constructed to be administered before the teaching/learning process. Their purpose is to provide information on students' placement at the outset of instruction, as well as to discover potential deficiencies or strengths in student learning.

The second category includes **mastery tests**. These tests are constructed for administration during the teaching\learning process. Their purpose is to provide a short term perspective on the successes of learning by describing how well students are progressing in their achievement of the lesson objectives; mastery tests also improve the learning process. Students' performance on the mastery tests are typically examined on an item by item (or cluster of items) basis. This information provides immediate diagnosis of each student's learning problems or success. Specific corrections and/or supplements to the original teaching are prescribed to correct these errors or enhance the enrichment learning path. The mastery test includes items for all the important elements in the lesson.

The third test category includes **unit (summative) tests**. These tests are administered periodically at the end of a series of lessons (representing the equivalent of an end of a "chapter" of a general subject, or an administrative period such as a quarter or semester.) These tests are used to gain information concerning the overall quality of the teaching\learning process and reflects achievement levels in a broader and more generalized perspective. These tests are also used for assigning grades and

reporting to parents or administrators. The level of generalization is the factor which most sharply differentiates intermediate units (summative) tests from formative mastery tests.

The fourth test category is the final (summative) test. This kind of test is constructed for administration after the end of the teaching learning process. The purpose is to provide a long term perspective on the success of the teaching process by describing the extent to which a student retained and transferred the expected learning towards which the program was targeted. Such information can be used for evaluating the teaching/learning process, for evaluating student progress over the entire period, and for evaluating the need for redesigning the instructional elements.

The diagnostic characteristics of the assessment component are represented in the following ways:

1. by using items which are diagnostic, i.e., each wrong response indicates a defined deficiency in learning (or the probable cause of the misunderstanding of the learned concept or skill).

2. by using tests which focus on skills and preferences that are relevant to successful learning and, thereby, provide information on students' potential and actual strengths and weaknesses.

3. by employing an assessment procedure at predetermined points along the learning process, where well defined sets of decisions regarding students' progress are made.

The diagnostic function of the system is rule based. The rules are determined by educational considerations regarding effective learning and instruction. These rules (specification of conditions and related prescriptions), when operated on student test peformance data, produce suggestions and recommendations for students' future learning.

IMPLEMENTATION

SERDS is currently implemented in several alpha-sites in the U.S.A. By the summer of 1989 SERDS instructional package will include 220 half-hour computer presented lessons, an Item Bank of some 3000 diagnostic questions, a multi-media resource guide and a collection of laboratory activities.

We hope that SERDS will demonstrate how a variety of technology based material and print material can be configured and utilized to provide students with an optimum science curriculum.

EDUCATIONAL NETWORKING/

CONSTITUTION D'UN RÉSEAU PÉDAGOGIQUE

ANALYSE DES BESOINS ET ÉVALUATION DES RÉSEAUX LOCAUX DE MICROORDINATEURS POUR LA CONCEPTION DE CENTRES DE RESSOURCES ÉDUCATIVES

A. Derycke, C. Vieville, D. Poisson
Laboratoire Trigone, Inst. C.U.E.E.P.
Université des sciences et techniques de Lille I
Villeneuve d'Ascq, France

L'interconnexion des microordinateurs constituant les stations de travail de l'apprenant est devenu une nécessité et ceci pour deux raisons essentielles: a) des contraintes d'ordre économique et logistique conduisant au partage de certaines ressources chères et à la création de bibliothèques de didacticiels centralisées; et b) l'apparition de besoins, à des fins éducatives, de moyens de communication et de coopération entre apprenants. Les réseaux locaux (L.A.N.) qui offrent une solution technique et matérielle performante, ont souvent été retenus.

Depuis quelques temps, des solutions "salles de classe", spécialement environnées pour la formation, sont apparues autour des stations de travail plus performantes (compatible PC ou équivalent). Nous avons mené une étude expérimentale très poussée des solutions proposées.

L'examen rapide des solutions adaptées à l'Éducation nous a montré qu'il existe un vide entre les solutions type "salles de classe" et les solutions type "campus" popularisées par des projets comme Athena (M.I.T.) ou Andrew (C.M.U.).

Les contraintes liées à la création de centres de ressources pédagogiques ont été précisées et permettent de mieux cerner les déficiences de ces propositions et de définir les développements de logiciels spécifiques visant à favoriser le principe "d'autonomie coopérative".

REQUIREMENT ANALYSIS AND EVALUATION OF LOCAL AREA NETWORK OF MICROCOMPUTER FOR THE SETTING UP OF LEARNING RESOURCE CENTRES

Inter-connection of microcomputers used as a learner workstation is becoming a necessity for two main reasons: a) constraints of economic or logistic orders which lead to share the experience devices or resources and to develop central courseware libraries; and b) emergency or needs, to educational purposes, in the fields of communication and cooperation between learners. The Local Area Networks (L.A.N.) have often been chosen because they offered a performant technical solution.

In a few cases a new classroom network has appeared, especially designed for the education domain and based on more powerful workstations (PC compatible). We have conducted experimental studies of these proposed solutions.

A rapid examination of the solutions adapted to the education have shown us that a gap exists between "classroom" solutions and "campus" solutions like those promoted by projects Athena (M.I.T.) or Andrew (C.M.U.).

The constraints relative to the setting up of Open Learning resources centres have been precise and they allow us to analyze drawbacks and deficiences of these solutions and to define specific requirements and needs of specialized software to manage such a center and to favour the principle of "cooperative autonomy".

ANALYSE DES BESOINS ET EVALUATION DES RESEAUX LOCAUX DE MICROORDINATEURS POUR LA CONCEPTION DE CENTRES DE RESSOURCES EDUCATIVES

A. DERYCKE - C. VIEVILLE - D. POISSON

Laboratoire TRIGONE - Institut CUEEP
Université des Sciences et Techniques de LILLE I
VILLENEUVE D'ASCQ - France

Le CUEEP, Centre Universitè Economie d'Education Permanente, Institut de l'Université de LILLE I, a trois missions : la formation initiale, notamment dans le domaine de l'Ingénierie de l'Education, la formation continue des adultes, la recherche. Le CUEEP occupe une place à part dans les universités françaises par sa taille en formation continue : plus de 80 000 heures de formation et plus de 12 000 inscrits chaque année, ainsi que par son intervention à tout niveau de formation (de l'illéttrisme au doctorat), et dans de nombreuses disciplines aussi variées que le français, l'anglais, les sciences, l'économie, l'informatique, la mécanique auto, ou l'audiovisuel et la psychosociologie...

Depuis déjà plus de dix ans, les technologies éducatives ont été introduites dans les formations d'adultes, en particulier les usages pédagogiques des microordinateurs (1). Ceci nous a amenés à produire des dispositifs matériels originaux (2) et de très nombreux didacticiels, dont un certain nombre fait l'objet d'une diffusion nationale.

I - VERS L'ENSEIGNEMENT OUVERT

I.1. L'évolution de la demande

Depuis de très nombreuses années, nous avons été amenés à mettre en place des filières de montée en qualification en formation générale pour des publics de bas nivau de qualification (avant le baccalauréat). Toutes les formations sont modulaires et offrent une validation par unités de valeur capitalisables. Habituellement, les rentrées sont semestrielles.

Même si ces dispositifs sont bien adaptés au public d'adultes, il est apparu depuis maintenant quelque temps, des difficultés liées à la demande des apprenants. En ce qui concerne les apprenants n'exerçant pas d'emploi salarié, et qui ne bénéficient pas d'un salaire de l'Etat pendant la formation, l'assiduité et les résultats finaux ont tendance à devenir plus faibles. Pour les publics salariés, en particulier au sein des formations intra-entreprises, il y a un début de rejet du stage groupé dans l'espace et dans le temps.

Les expériences que nous menons depuis maintenant deux ans, nous montrent qu'il faut aller vers plus d'individualisation en termes d'objectifs et de parcours, vers plus de souplesse en terme de gestion des cursus, des lieux d'apprentissage.

I.2. L'enseignement ouvert

Notre définition de l'enseignement ouvert est : "offrir une plus grande liberté de choix à l'apprenant quant aux lieu, durée, rythme et stratégie d'apprentissage, ou encore domaine d'intérêt de la formation".

C'est effectivement très proche du concept d'"Open Learning" de nos collègues anglo-saxons. Mais il faut savoir que des traditions culturelles et des contraintes légales et financières ne facilitent pas la généralisation d'une telle approche en France. De simples considérations économiques nous montrent que cette approche n'est possible en formation d'adultes qu'à deux conditions :

- aider, guider l'apprenant face à ce libre choix et repenser les mécanismes de validation ;
- faire appel à l'autoformation des apprenants pour une partie plus ou moins importante de leur cursus.

I.3. Un continuum de modes éducatifs et de nouvelles fonctions

La souplesse, ainsi que la possibilité d'une évolution progressive, nous ont conduits à mettre en place un continuum (au sens de sans rupture), allant de la formation en groupe individualisé à l'enseignement à distance, en passant par l'autoformation assistée par tuteur, en self service et utilisant des centres de ressources éducatives.

Le groupe individualisé signifie qu'il y a encore unité de lieu et de temps, souvent de disciplines étudiées (ex : les mathématiques) mais que, par le fonctionnement en ateliers de quelques personnes ou en utilisant les Nouvelles Technologies Educatives (NTE), il y a possibilité de stratégies et d'évolutions différenciées.

A l'opposé, dans l'enseignement à distance, la plus grande partie de la formation est en autoapprentissage, et le tutorat/guidance est assez peu développé.

Ces différents modes d'éducation nous conduisent progressivement à la création de nouvelles fonctions (ou à l'évolution de celles-ci lorsqu'elles existent), telles que le tuteur, le conseiller en formation, ou encore le rôle du formateur matière.

I.4. Apport des NTE et création des centres de ressources éducatives

Cette évolution est inconcevable sans un usage important des NTE, en particulier de l'EAO, et de l'existence de supports pédagogiques multimédias : les documents papiers, les didacticiels, les cassettes vidéo ou audio, la télématique. Dans notre cas, les NTE sont vues comme des auxiliaires précieux, tant pour l'individualisation de la formation (le respect des rythmes) que comme amplificateur/facilitateur dans la formation en groupe individualisé.

Au fil des années, notre parc d'équipement est devenu important : plus de 200 stations de travail EAO. Notre médiathèque comporte aujourd'hui plus de 400 didacticiels conçus par nous, de très nombreuses cassettes audio et vidéo.

Afin d'être plus proches des adultes en formation, nous possédons neuf centres de formation répartis sur la région (4 millions d'habitants). C'est pourquoi, pour conserver cette proximité, nous avons décidé la mise en place progressive de centres universitaires de ressources éducatives : les ressources étant à prendre en terme de ressources humaines (tuteur, conseiller, formateur), de ressources matériels (poste de travail) ou de ressources médiatiques.

II - LES CENTRES DE RESSOURCES EDUCATIVES : ANALYSE DES BESOINS ET CONTRAINTES

Nous pouvons adopter quatre points suivant les acteurs considérés : apprenant, tuteur, enseignant, gestionnaire. C'est ce que nous avons fait lors d'une analyse organique et fonctionnelle, afin de déterminer la meilleure architecture matérielle, logicielle et organisationnelle d'un centre de ressources.

II.1. Le point de vue de l'apprenant

L'introduction de modes d'apprentissage plus individualisés, plus souples, introduit la nécessité absolue de mettre en place un dispositif d'accueil, de bilan, de tuteurage, de conseil, d'évaluation, performant.

Du point de vue technique et étant donné la population utilisatrice de chaque centre (hypothèse : entre 300 et 500 personnes par semaine), il faut prendre en compte très sérieusement la dimension du suivi individualisé. Chaque apprenant doit pouvoir conserver une trace de ces activités (sur disquette, serveur d'administration, plus tard une carte à mémoire). De la même façon, que ce soit en centre de ressources ou en enseignement à distance, il est souvent souhaitable de pouvoir imprimer des résultats issus de didacticiels. Ceci entraîne la création d'ateliers d'impression.

- Le point de vue du tuteur/conseiller

Celui-ci veut pouvoir traiter les "traces" du suivi d'un apprenant (visualisation du parcours, résultats d'évaluation, temps passé sur chaque média, ressources éducatives déjà utilisées). Il souhaite également être assisté d'un outil informatique permettant une meilleure gestion des places disponibles dans le centre de ressources et de la médiathèque. Ceci impose la constitution de bases de données partagées.

- Le point de vue des enseignants/pédagogues

Au sein d'une discipline, les enseignants souhaitent pouvoir enrichir, modifier, adapter les supports pédagogiques multimédias. Il faut donc qu'ils aient accès également à la gestion de la médiathèque. Les enseignants ont deux types d'activités : celles qui ont trait à la formation en groupe, et celles qui correspondent au soutien pédagogique des apprenants. Dans ce cas, ils souhaitent pouvoir accéder au système de suivi des apprenants, et mettre en place des moyens de communication asynchrones (cahier de suivi ou mieux, une messagerie électronique). Enfin, le contrôle des connaissances impose la gestion de banques d'exercices et d'examens.

- Le point de vue du gestionnaire

Il faut prendre en considération les aspects légaux de la formation d'adultes : quel est le référentiel pour la subvention accordée pour une formation d'adultes qui ne soit pas réalisé en stage classique ?

Il faut donc, là également, un suivi des temps passés à chaque activité, des niveaux atteints, etc.

De même, la plupart des formateurs et tuteurs de la formation continue sont rémunérés, en France, à la vacation. Un suivi des activités est donc également nécessaire.

Enfin, il faut prendre en considération des problèmes tels que les droits des apprenants (accessiblité à telle ou telle formation, durée...) et ceux liés à la protection des ressources matériels et médiatiques.

III - UNE ANALYSE RAPIDE DES SOLUTIONS EXISTANTES

La nécessité de communiquer des informations entre les différents acteurs et d'accéder à des ressources partagées (serveur de didacticiels, par exemple), impose la mise en place d'un système d'information basé sur des réseaux locaux de microordinateurs (3).

A l'heure actuelle, dans le système éducatif, il existe deux types de solutions : les réseaux locaux de salles de classe et les réseaux locaux de Campus.

III.1. Les réseaux locaux de salle de classe

Hormis les solutions que nous avons développées nous-mêmes (2), il existe d'autres propositions commerciales comme Classnet d'Olivetti en Europe (4) ou le Icon-System (5).

La principale caractéristique de ces solutions est de supporter, en général, un nombre assez faible de stations de travail élèves. La topologie est simple et les dimensions du réseau local limitées à la salle de classe.

Par rapport aux réseaux locaux utilisés en bureautique, la valeur ajoutée réside souvent dans la fourniture d'un logiciel de gestion de la banque de didacticiels (catalogage). Mais celui-ci est, en général, assez frustre et fermé. Il n'offre que très rarement un suivi des accès aux ressources par les apprenants (durée, quand ?).

Par contre, des outils sont offerts pour faciliter l'usage collectif (tableau noir électronique -3- -4-), la prise de contrôle à distance d'une station élève par un tuteur permettant ainsi l'assistance "en ligne" ou encore la messagerie électronique en temps réel (5).

III.2. Les réseaux locaux de Campus

Ils correspondent, la plupart, à des projets très ambitieux popularisés par des projets comme Athéna au MIT ou Andrew à Carnegie Mellon.

La caractéristique générale de ces réseaux est d'être très proche des réseaux de la recherche et de servir d'objets à la recherche : station de travail haut de gamme (Sun, PC/RT...), réseaux locaux de type ingénierie (ethernet, large bande), existence d'un centre de calcul interne au Campus avec un support technique important. Les usages sont plutôt tournés vers la simulation, le calcul numérique, la bureautique scientifique, la programmation.

La plupart des logiciels sont développés par les départements informatiques de ces universités et n'ont pas encore atteint le stade commercial, sauf peut-être Andrew (IBM).

L'examen de ces deux types de solutions montre qu'elles sont toutes les deux inadaptées à la réalisation de centres de ressources éducatives , tels que nous les avons définis précédemment.

Nous avons donc choisi une démarche très pragmatique vers la recherche d'une solution idéale, suivant en cela une démarche voisine de celle de l'Université de WATERLOO (6).

IV - LA SOLUTION EN COURS DE DEVELOPPEMENT

D'un point de vue architecture matériel, nous avons décidé d'adopter à chaque fois que c'est possible, des équipements standards :

- Les stations élèves et tuteurs sont constitués de PC/AT (1 MO de mémoire - Ecran VGA).
- Les serveurs de fichiers, d'impression, de communication, sont constitués de PC/AT 286 ou 386 suivant la taille du centre de ressources.
- Le standard adopté pour le réseau local est le IEEE 802/3-Starlan, ce qui permet un système de câblage souple et assez peu onéreux : utilisation du câble téléphonique.

Notre architecture est modulaire : un module doit pouvoir accueillir 100 à 150 personnes/semaine (hypothèse moyenne : 4 H/semaine). Il comporte huit postes de travail informatique : deux postes pour la vidéo (magnétoscope ou vidéodisque), deux postes pour le travail audio-comparatif. Un centre de ressources comprend, en moyenne, trois modules auxquels il faut associer un module technique (serveurs d'impression, de didacticiels, de communication), avec un terminal d'administration, ainsi qu'un module d'accueil (tuteur) et un module formateur.

La valeur ajoutée à ce système, qui utilise des configurations habituellement rencontrées en bureautique, réside dans la création d'une structure d'accueil sur les stations élèves rendant complètement transparent l'usage de la didacthèque, ainsi que d'applications autour d'une base de données pour la gestion du suivi des apprenants et des ressources utilisées.

Notre expérience montre, qu'en matière de logiciels commercialisés pour la réalisation de réseaux locaux (pour la gestion de serveurs de fichiers en particulier), il ne faut pas hésiter à se tourner vers des solutions haut de gamme plus onéreuses (système Novell SFT ou Vines de Banyan).

Dès que le nombre de stations devient important (> 20), les performances sont meilleures que pour le système simple (MS/NET). L'investissement en serveurs de fichiers est donc moindre et les gains ainsi effectués compensent largement le prix de ces logiciels. De plus, ils offrent des fonctionnalités avancées telles que la gestion du profil des utilisateurs (droit d'accès, durée, etc) absolument indispensable à la gestion pédagogique et administrative d'un centre de ressources.

VI - VERS L' "AUTONOMIE COOPERATIVE"

Le concept d' "autonomie coopérative" que nous avons déjà proposé, et qui nous a servi de principe directeur à la réalisation du Nanoréseau (réseau local de salle de classe) (2), peut être défini comme :

- Autonomie : pour l'individualisation, la maîtrise personnelle d'une station de travail...

- Coopérative : pour le partage des ressources, mais également la réintroduction d'une dimension collective dans l'autoformation. Ceci nécessite la mise à disposition d'outils de dialogue/conversation entre les différents acteurs. Pour cela, nous avons distingué deux types de communication :
. Les communications asynchrones (sans connexion entre les acteurs) qui se relaient sur des systèmes de stockages magnétiques ou optiques. Les systèmes de Téléconférences Assistées par Ordinateurs peuvent être adaptés à l'enseignement et permettre la création de "salles de classe virtuelles" (6).
. Les communications synchrones, équivalentes à une conversation en face à face (avec connexion).

Nous avons mis en chantier un projet de recherche visant à réaliser un système de conférence temps réel multimédia (voix + données + images numériques) pour le télé-tuteurage, que ce soit au sein des centres de ressources en s'appuyant sur les réseaux locaux (LAN) ou à distance en s'appuyant sur le développement rapide en France de réseaux RNIS (ISDN).

REFERENCES

1. A. DERYCKE - J. LOSFELD
"Impact of new learning technologies on an academic institution and its environment". Proceeding of the ECCE 88 - LAUSANNE - IFIP - Juillet 88 - Ed. F. LOVIS, E. D. TAGG - North Holland - pp 449-454.

2. A. DERYCKE - C. VIEVILLE - D. POISSON - C. STACH - N. N'GUYEN
"Le Nanoréseau : utilisations pédagogiques d'un réseau local" - TSI Techniques et sciences informatiques - janvier 88 - pp 13-26 - Dunod éditeur

3. A. DERYCKE - P. LOOSFELT - C. VIEVILLE
"Toward a pedagogy in networks".
Proceeding of Iccal 87 - University of CALGARY ALBERTA - May 87 - pp 93-98

4. P. FIARANI GALLOTA - G. GAZZANO - F. PIMPINELLI
"Distributed Environment for Education : the electronic blackboard" - Cinquième Symposium canadien sur la technologie pédagogique - OTTAWA - Canada - 5 au 7 mai 1986 - pp 577-583

5. R. S. MC LEAN
"Local Area Networks in Education : an infrastructure perspective" - Computer in Education - F. LOVIS - ED TAGG - IFIP 88 - Elsevier Science publishers BN (North Holland).

6. D. D. COWAN - S. FENTON - J. N. GRAHAM - T. M. STEPIEN
"Networks for Education at the University of WATERLOO".
computer Network and ISDN systems : N, 15 - 1988 - pp 313-327

LA COMMUNICATION ASSISTÉE PAR ORDINATEUR DANS L'ENTREPRISE. VERS UNE STRATÉGIE DE FORMATION EN COURS D'EMPLOI ET DE PERFECTIONNEMENT CONTINU

F. Henri, J.-Y. Lescop
Télé-université
Université du Québec
Montréal, PQ

Après la période fébrile où les entreprises se sont dotées d'équipement de bureautique pour améliorer leur productivité, il reste maintenant a développer les savoir-faire et les méthodes de travail qui permettent au personnel du secteur tertiaire de les exploiter rationnellement.

La présente recherche a pour but de mettre en évidence les nouveaux paradigmes de la formation assistée par ordinateur par l'intermédiaire d'un modèle opérationnel basé sur l'utilisation de la télématique à des fins de formation et de perfectionnement dans l'entreprise.

Le modèle, élaboré à partir des plus récentes recherches en matière d'implantation d'innovation et d'utilisation de la téléconférence assistée par ordinateur comme moyen de formation, devrait permettre aux formateurs dans les entreprises de développer des programmes qui, par la forme et le contenu, visent l'adaptation professionnelle aux nouvelles technologies.

COMPUTER-ASSISTED COMMUNICATION IN THE BUSINESS SECTOR. TOWARDS A STRATEGY FOR ON-THE-JOB TRAINING AND CONTINUING DEVELOPMENT

Now that we have passed the period of feverish activity during which corporations acquired office automation equipment to improve their productivity, we must develop the skills and work methods that will allow employees in the tertiary sector to use this equipment in a rational way.

The purpose of this research is to discover the new paradigms for computer-assisted training through an operational model based on the use of telematics for training and development in business.

The model, developed from the most recent research in the application, innovation and use of computer-assisted teleconferencing as a training tool, should enable the training personnel of companies to develop programs whose form and content are aimed at occupational adaptation to new technologies.

LA COMMUNICATION ASSISTEE PAR ORDINATEUR DANS L'ENTREPRISE
Vers une stratégie de formation en cours d'emploi et de perfectionnement continu

F. Henri, J-Y. Lescop

Télé-université, Université du Québec

La recherche nous a permis d'analyser différents modèles d'utilisation de la téléconférence assistée par ordinateur (TAO) et d'élaborer un modèle conceptuel de formation à distance intégrant la TAO. Elle prévoyait également l'opérationnalisation d'une grille d'analyse de messages, des consignes de traitement des messages, des consignes d'animation des communications assistées par ordinateur.

A la suite d'une pré-expérimentation menée au printemps 1988 et de l'expérimentation de l'automne auprès d'agents-conseil du Mouvement Desjardins, nous avons relevé quatre facteurs importants qui méritent d'être étudiés en profondeur afin de parfaire et de compléter notre modèle; il s'agit de facteurs liés:

1- à l'organisation (infrastructure de support à la formation, ressources pédagogiques, sécurité et protection des données);

2- aux habiletés des utilisateurs à se servir du médium TAO;

3- au développement des habiletés cognitives des participants;

4- à l'appropriation du processus de résolution de problèmes par la téléconférence assistée par ordinateur (TAO).

La recherche nous a permis en outre d'opérationnaliser et de valider ces facteurs et variables qui s'ajoutent au modèle en:

A) procédant à une étude d'impacts pour:

a) mesurer le degré de satisfaction des gestionnaires de la formation sur l'utilisation de la téléconférence assistée par ordinateur (TAO);

b) identifier les facteurs qui affectent l'implantation de la TAO dans le processus de la formation;

c) gérer les changements qui affectent l'organisation ainsi que l'approche et la démarche de la formation (conception, production, évaluation);

B) mesurant le degré d'habileté requis pour utiliser efficacement la téléconférence assistée par ordinateur (TAO) par les agents-conseil;

C) concevant une grille d'analyse des activités cognitives pour fournir au formateur un outil fiable et fidèle de support à l'apprentissage;

Introduction

L'introduction du traitement de texte, du chiffrier électronique, des bases de données, l'accès au courrier électronique et aux conférences assistées par ordinateur via la communication télématique, entrainent une profonde transformation du travail. Les recherches de Hiltz[1] ont permis de cerner les transformations provoquées par l'informatisation du travail et de les répertorier en quatre grandes catégories:

- le décloisonnement des perspectives et des activités professionnelles,

---

[1]Hiltz, Roxane. (1985) On Line Communities: A Case Study of the Office of the Future. Ablex Publishing Corporation. New Jersey.

- l'accroissement de la fonction de communication,
- la transformation des notions de temps et d'espace
- le changement des habitudes de travail (intensification du rythme de travail, meilleure organisation, travail à la maison,...)

Cette transformation qualitative du travail requiert une période d'adaptation professionnelle qui n'est pas uniquement à la charge de l'employé. L'employeur détient une responsabilité importante quant à la formation et au perfectionnement de son personnel dans ce domaine.

Or l'intégration des différents outils de bureautique dans l'entreprise et plus particulièrement la téléconférence assistée par ordinateur, sont susceptibles de transformer l'approche traditionnelle de la formation en cours d'emploi. Les résultats de recherche sur l'utilisation des nouvelles technologies dans des contextes de formation (Bates[2], Harasim[3]) démontrent les nombreux avantages de la téléconférence assistée par ordinateur comme moyen d'apprentissage.

Objectifs

Le projet doit nous permettre principalement d'analyser différents modèles d'utilisation de la téléconférence assistée par ordinateur (TAO) et d'élaborer un modèle conceptuel de formation intégrant la téléconférence assistée par ordinateur. Il prévoit également l'opérationnalisation de variables telles une grille d'analyse de messages, des consignes de traitement des messages, des consignes d'animation des communications assistées par ordinateur, etc...

A la suite de la pré-expérimentation menée de mai à août 1988 et de l'expérimentation en cours, nous avons relevé quatre facteurs importants qui méritent d'être étudiés en profondeur afin de parfaire et de compléter notre modèle; il s'agit de facteurs:

*liés à l'organisation (infrastructure de support à la formation, ressources pédagogiques, sécurité et protection des données);

*liés aux habiletés des utilisateurs à se servir du médium TAO;

*liés au développement des habiletés cognitives des participants;

*liés à l'appropriation du processus de résolution de problèmes par la téléconférence assistée par ordinateur (TAO).

Nous verrons ensuite à opérationnaliser et de valider ces facteurs et variables qui s'ajoutent au modèle en:

1-procédant à une étude d'impacts pour:

a)mesurer le degré de satisfaction des gestionnaires de la formation sur l'utilisation de la téléconférence assistée par ordinateur (TAO);

b)identifier les facteurs qui affectent l'implantation de la TAO dans le processus de la formation;

c)gérer les changements qui affectent l'organisation ainsi que l'approche et la démarche de la formation (conception, production, évaluation);

---

[2]Bates, A.W. (1986) "Which Way For Information Technology in Distance Education"." in Revue de l'enseignement à distance. Vol.1, No 1.

[3]Harasim, Linda. (1987). "Teaching and Learning On-Line: Issues in Computer-Mediated Graduate Courses" in Canadian Journal of Educational Communication. Vol 16, No 2.

2-mesurant le degré d'habileté requis pour utiliser efficacement la téléconférence assistée par ordinateur (TAO) par les agents-conseil;

3-concevant une grille d'analyse des activités cognitives pour fournir au formateur un outil fiable et fidèle de support à l'apprentissage;

Hypothèses

A partir des recherches portant sur les téléconférences assistées par ordinateur (Hiltz et Turoff[4], Haile et Richards[5]), il est possible de formuler des hypothèses sur l'effet de certains facteurs sur l'efficacité de la téléconférence comme mode de perfectionnement en cours d'emploi.

Hypothèse générale

L'introduction de la téléconférence assistée par ordinateur (TAO) dans le cadre d'un programme de perfectionnement offert aux agents-conseil financiers du Mouvement Desjardins sur les nouveaux services d'aide à la retraite, favorisera l'apprentissage.

Hypothèses spécifiques

Hypothèse 1: effets des attitudes sur l'utilisation des téléconférences assistées par ordinateur.

1.1 Les agents-conseil financiers qui auront des attitudes positives par rapport à la téléconférence assistée par ordinateur, l'utiliseront davantage que celles qui auront des attitudes négatives.

1.2 Les agents-conseil financiers qui auront des attitudes positives par rapport à la téléconférence assistée par ordinateur, exprimeront un plus haut niveau de satisfaction par rapport au système et l'évalueront de façon plus positive; on trouvera des corrélations positives entre le niveau de satisfaction, l'évaluation du système et les variables reliées à l'utilisation du système.

Hypothèse 2: effet du niveau de littéracie sur l'utilisation de la téléconférence assistée par ordinateur.

2.1 Les agents-conseil qui auront un plus haut niveau de littéracie tel que mesuré par leurs habitudes, leurs attitudes, leurs comportements et leurs habiletés par rapport à l'écrit, utiliseront davantage la téléconférence assistée par ordinateur, produiront davantage de messages et communiqueront de façon plus efficace.

Hypothèse 3: effets de la téléconférence assistée par ordinateur sur l'apprentissage.

3.1 L'utilisation de la téléconférence assistée par ordinateur contribuera à développer les processus de compréhension de l'information transmises dans le cadre du programme de perfectionnement.

3.2 L'utilisation de la téléconférence assistée par ordinateur facilitera l'apprentisaage des processus de résolution de problèmes nécessaires à la réalisation de la tâche de conseiller en programme de préparation à la retraite (programme de formation choisi dans le cadre de l'expérimentation.)

3.3 L'utilisation de la téléconférence assistée par ordinateur contribuera à développer des habiletés cognitives requises pour conseiller les clients dans l'utilisation des services de préparation à la retraite.

---

[4]Hiltz, R., Turoff, M. (1978).- The Network Nation. Human Communication via Computer. Addison-Wesley, Mass.

[5]Haile, P., Richards, A.J. (1984) Supporting the Distance Learner with Computer Conferencing. New York Institute of Technology. New York.

Méthodologie

Trois types de variables interviennent dans ce projet: les variables indépendantes, les variables contrôles et les variables dépendantes.

Parmi les variables indépendantes, on trouve celles liées au médium (médium écrit, médium TAO), liées aux sujets (attitudes par rapport au médium, niveau de litéracie, habiletés en compréhension et en production de texte), et celles liées à la tâche (méthodes traditionnelles de formation, méthode combinée, méthode TAO) et au contenu d'apprentissage.

Parmi les variables contrôles, on trouve celles contrôlées par les chercheurs (temps d'apprentissage, ordre d'apprentissage des contenus) et celles sur lesquelles les chercheurs ne peuvent intervenir (médium, sujets, grandeur de l'échantillon, niveau de scolarité, matériel de formation développé par le Mouvement Desjardins).

Enfin, parmi les variables dépendantes, on retrouve celles qui concernent l'utilisation du système (fréquence d'utilisation, durée moyenne par séance, nombre de messages produits et reçus), la compréhension, l'efficacité de la production écrite et de la communication via la téléconférence assistée par ordinateur (TAO),l'apprentissage du contenu du cours, le niveau de satisfaction par rapport au système et l'évaluation du système de téléconférence assistée par ordinateur.

Déroulement des activités additionnelles

L'expérimentation proprement-dite se déroule sur une période de six semaines, (du 19 octobre au 2 décembre 1988) dans le cadre d'un cours préparé par le Mouvement Desjardins portant sur la préparation financière de la retraite. Ce cours est divisé en quatre parties d'une semaine chacune au cours desquelles les participants pourront discuter via la TAO, lors des semaines deux et quatre seulement, des solutions aux problèmes qui leur seront posés. Des animateurs et experts interviendront alors sur le système pour aider les participants dans leur démarche de résolution de problèmes.

Une analyse exhaustive des messages (protocoles) permettra de vérifier la présence et le rôle des variables. Lors des deux autres semaines, les participants devront discuter de leur solution aux problèmes directement avec un expert via le téléphone. Un pré et un post tests seront administrés à tous les participants pour mettre en évidence certaines caractéristiques des individus quant aux variables énoncées plus haut.

Impacts du projet

Les avantages pour les organisations d'implanter et d'intégrer la téléconférence assistée par ordinateur dans le processus de formation et de perfectionnement continus se situent à plus d'un plan:

1-Augmentation de la qualité de la formation: échanges nombreux et formateurs entre les participants, les experts conseil et les formateurs; accès à des banques d'informations numérisées; nouvelle dynamique de formation fondée sur la participation active du personnel à former.

2-Rentabilisation des équipements de bureautique grâce à une rationnalisation des coûts de déplacements, à la diminution des absences du personnel en formation et à l'intégration de ces nouvelles technologies éducatives au processus de formation et de perfectionnement continu.

3-Développement du sentiment d'appartenance des employés dispersés à travers la province (Région de Québec, du Saguenay-Lac St-Jean, du Bas Saint-Laurent et de la Gaspésie) au Mouvement Desjardins.

CO-CO IN NURSING CARE PLANNING: AN INNOVATIVE APPROACH TO STUDENT LEARNING

B.J. Sleightholm Cairns
Faculty of Nursing
University of Western Ontario
London, ON

P.J. O'Kelly
Faculty of Education
University of Western Ontario
London, ON

This demonstration field study evaluated the use of an innovative computer conferencing system to enhance student nurses learning of the nursing process through care planning. Through the use of a computer conferencing system (CO-CO) developed by an educational software company, the researchers were able to set up a learning situation which would allow the students the chance to practise care planning interactively. This care planning could be conducted independently of the teacher on a preset case study, allowing the teacher to engage in other activities while still giving her access to the group for essential feedback. The particular software package, CO-CO (computer conferencing) facilitates real-time, synchronous communication among students. The researchers used a convenience sample of eight students who had been grouped together as a clinical group. This study offers an alternative approach to the teaching of nursing care planning and opens the door for future research using this computer conferencing system.

CO-CO UTILISÉ DANS LA PLANIFICATION DES SOINS INFIRMIERS : UNE NOUVELLE FAÇON D'ENSEIGNER

Cette étude d'une démonstration pratique a permis d'apprécier la valeur d'un système de conférence par ordinateur comme outil pour améliorer l'enseignement du nursing, en prenant comme sujet la planification des soins infirmiers. Les chercheurs, au moyen d'un système de conférence par ordinateur (CO-CO) mis au point par une société de logiciels éducatifs, ont pu définir une situation d'apprentissage qui donne la chance à des étudiants de pratiquer interactivement le processus de planification des soins. Cette planification pouvait être conduite indépendamment du professeur à partir d'une étude de cas prédéterminée, libérant ainsi ce dernier pour d'autres tâches tout en lui laissant la possibilité de donner au groupe des indications essentielles. Le progiciel particulier utilisé, CO-CO (COmputer COnferencing) facilite le dialogue simultané et en temps réel entre les étudiants. Les chercheurs ont constitué un échantillon commode de huit étudiants qu'ils ont organisé comme un groupe clinique. Cette étude donne les résultats d'une démarche innovatrice pour l'enseignement de la planification des soins infirmiers et ouvre la voie à d'autres recherches sur l'utilisation des systèmes de conférence par ordinateur.

CO-CO IN NURSING CARE PLANNING: AN INNOVATIVE APPROACH TO STUDENT LEARNING

B.J. Sleighthole Cairns, P.J. O'Kelly

University of Western Ontario
Groupware Technologies Corporation

This qualitative study investigated the use of an innovative computer conferencing system to enhance student nurses learning of the nursing process through nursing care planning. In nursing care planning, nurses establish strategies to meet a patient's needs. Through assessment of the patient, appraisal of the subjective and objective data, and establishment at a nursing diagnosis or problem statement, nurses select interventions and determine evaluation criteria. Student nurses need to be able to problem-solve care for their patients rather than to simply memorize standards of care from textbooks.

Traditionally nurses learn nursing care planning in the classroom by the case study method which is then reinforced in the clinical area. However this reinforcement requires many hours of work for the student and for the instructer to evaluate and provide feedback. Since care planning does require practise in more than one patient situation, it is not a skill that can be taught by lecture method only.

As faculties of nursing experience increasing budgetary constraints while maintaining their present student enrolment, nursing faculties face the need to teach essential skills such as care planning in a more time-efficient, cost-effective manner. Through the use of a computer conferencing system developed by a small educational software company the researchers set up a learning situation which allowed the students to practise their care planning interactively. This care planning could be conducted independently of the teacher, thus allowing the teacher to work in other academic endeavours. At her convenience the teacher could monitor the student group interaction and provide appropriate feedback.

Review of the Literature

The literature of CAI cites many advantages to the use of computer assisted learning instruction. The computer has been used in a number of different manners to teach many different skills in nursing. Four major types of CAI exist in nursing: problem solving, drill and practise, tutorials and simulations (Hassett,1984). The computer is said to be infinitely patient with students, providing students with immediate feedback and encouraging their idependence and motivation (Bitzer & Bitzer 1973; Hassett, 1984).

Walker (1986) in her national survey of deans and chairpersons of schools of nursing in the United States concerning the use of computers, indicated limited use of all the computer's capabilities. Word processing and administrative functions were primary uses with limited research and instructional applications.

Dr. Gordon McCalla in an address delivered at the International Conference on Computer Assisted Learning stated some of the difficulties of computer assisted learning. Traditional approaches to computer assisted instruction tend to be "rigid, unresponsive and dictatorial" (McCalla, 1987, p.32). As technologies improve, the software has noticeably improved but the rigidities in many of the programs remain.

The main rigidity is that the computer has been unable, thus far, to deal with natural language and does not allow the student to answer in unpredictable ways . Blowmire (1988) has suggested that software such as the CO-CO conferencing program represents a new paradigm for computer assisted learning in the classroom because it puts real intelligence at the disposal of the users, allowing them to interact intelligently across the network using natural language.

CAI traditionally has allowed one student to interact with a single computer using a single program. It is based on a two-way interaction between a computer and a learner by the use of an operating system and a software program (Saba and McCormick, 1986). Through the software package, content is presented and the student

responds to that content in some pre-determined format. The software available has not allowed for student-to-student interaction nor natural language problem-solving through the use of the computer.

Beckwith (1987) discussed group problem-solving via computer conferencing as the computer's most promising potential. Computer conferencing has the capability of bringing individuals together for a variety of "socially worthwhile interaction to purposeful change through group problem-solving" (Beckwith, 1987,p.89).

New communications tools, such a CO-CO, help realize this potential by making it possible for students to communicate and problem-solve in groups. Dr. Jo Ann Wilton (1987) in her report Evaluation of the Software Package,CO-CO, discusses the use of CO-CO, as a "facilitator of interaction"(1987,p.13). She describes it as not being a typical software package but rather a utility which facilitates real-time communication among students. Since CO-CO is devoid of content, its usefulness can be stated in terms of processes encouraged rather than any particular content. Although Dr. Wilton's report deals with a field study conducted in five sites in the elementary and secondary school system, its application in other educational systems is evident.

The researchers, after reviewing the literature,realized the potential of this software package for nursing education. Instead of using CAI to teach traditional skills whereby the student is interacting with the computer, CO-CO created an environment which not only facilitated student interaction but actually seemed to stimulate it.

Methodology

The population studied were student nurses in the first of a four-year, under-graduate program. The researchers used a convenience sample of eight students who had been grouped together as a clinical group since the beginning of their first year.

The students participated in two sessions. The first session, provided students with two-hours training. They became familiar with the Icon microcomputer and learned how to operate CO-CO — the computer conferencing software program. The students were randomly assigned to work in pairs at four different workstations. The researchers were located at another work station. After an initial computer demonstration, the researchers and the students began interacting on the screen with CO-CO. Simple conversation continued until the students appeared comfortable with the interaction that was taking place on screen.

The CO-CO communications screen is divided into six segments, one for each workstation and a sixth (data base) in which the researchers displayed a patient case-study. After the students appeared more comfortable with the software's capabilities, the students' attention was directed to the "data base" screen which in included the patient case study and a a sequence of simple tasks which required the students to engage in problem-solving processes.

The training session served a multitude of purposes. Since the software had never been used in such a manner, it allowed the researchers an opportunity to try out different possibilities for its use. It also allowed time for the students to become familiar with the equipment and the software. The use of a simple case study allowed them the chance to try to use the software on a legitimate task rather than just interacting without the benefit of a directed activity.

The second session, which occurred a week later, was a more structured experience. The researchers had once again prepared the CO-CO problem-solving activity in advance. The students were given a few minutes to refamiliarize themselves with the setup and then could begin work. The tasks outlined for this session were designed to encourage a higher degree of interaction than had occurred in the first training session.

The tasks divided the responsibility for a collective care plan into four stages. Each team of students was given responsibility for developing a different portion of the overall group nursing care plan. Thus, the group in charge of creating the initial portion of the plan was able to request and receive feedback simultaneously from the other groups who played a supportive, consultative role. Once consensus had been achieved on the first task, another group took over and the process began again -- this time, the members of the first group became part of the supportive/consultative team.

This session lasted about two hours. Initially, through the use of CO-CO, the students brainstormed about the task at hand and organized the data concerning the patient presented. Two students assumed the leadership for the assessment phase information, questionning when appropriate via the screen Following this phase, the next task was to collaboratively arrive at a suitable nursing diagnosis. Through much discussion on screen, they reached a nursing diagnosis which could correctly focus the planning, implementation and evaluation phases.

At each phase of the nursing process, different student leaders appeared to emerge. When disagreements arose, the students had to justify their decision and the conflict was resolved as new information surfaced. During the two hour time, the students collaboratively arrived at a workable care plan for the patient outlined in the case study. Throughout this interactive experience the researchers interacted with the group as consultants.

Following both sessions the students made positive statements about the experience. They felt the sessions were very helpful in learning more about care planning. The reasons cited included the ease with which they could use the software CO-CO to interact and reach decisions about care. They could also learn from each other, and address each other's errors in care planning because they could see the live interaction on the screen.

After the sessions, the students completed a questionnaire. Face and content validity of the questionnaire was established by allowing the developer of the software and a nursing instructor to read and make comments. It is an open-ended questionnaire allowing the nursing students to respond in their own words to the questions about learning on the computer.

Findings

All eight students involved responded to the questionaire. Six students had some previous experience in using computers in a learning situation. However some of the experiences were quite limited and not very positive in nature. One student had stated that she had felt quite intimidated by the "computer whizzes" in the class at high school and that had affected her learning. Two of the students had taken a data processing course in high school.

When asked about their expectations concerning the use of computers in this particular learning experience in nursing, the responses varied from curiousity to apprehension and reluctance. "I was reluctant about combining nursing and the computer. I wasn't sure how using the computer would assist us in learning nursing process." "I was curious to see how efficient a tool would be in formulating a care plan with 8 people." "I really didn't go for the idea at all, as I hated computers."

Considering that the key questions deal with the learning which took place in using the software, the responses to those questions will be considered in some detail. Six of the eight students felt their understanding of nursing care planning was enhanced after their computer sessions. Their explanations included comments such as:

> ...nursing care planning could be looked at in a broader sense. Rather than just one individual working,the entire group could work together to problem-solve...
>
> ... In encountering a problem we could find the solution effectively together rather than by just flipping through books and eventually saying, forget it...
>
> ...I never relized before how each step of the plan was built on the previous stage. Practical orientations were made more apparent...
>
> ...No one had to stop and take notes. Everyone got a chance to talk...

Those students who said their learning was not enhanced by the experience cited the fact that they had already made a few care plans and already understood how to do it.

One of questionnaire questions asked if the experience was enjoyable. The response was overwhelmingly positive. The experience was: "fun, it was a new way of learning". "I wasn't intimidated anymore". "I dion't feel pressured about the task - the simplicity of running CO-CO made this experience a fun one!" "I'd do more care planning because it's more efficient."

All students concluded that they had gained new insights into care planning through this experience. The group could interact, share their ideas easily and reach solutions. They saw the group as a helpful resource. One partipant commented: "I had to be more organized because others in the group would be using my information."

Another key component of the learning was that, through the use of CO-CO, the students learned to work more effectively as a group. They cited that the interaction and exchange of ideas and constructive criticism were benefits to their group. The product - the nursing care plan - was a better one because of the interaction. One student said that, in verbal interaction, not everyone gets the chance to talk, while everyone contributed to this interaction. None of the students in the group felt knowledge of computers or lack of such affected her ability to use the system.

The possible problems the students cited in using CO-CO in other learning situations included: concerns that some students may contribute more than others, if the groups were too large the screen would be confusing and the amount of screen space would be insufficient.

When asked about the independence of the

task and the need for continuous teacher supervision, the comments varied. Students reported they could work independently as a group. However the degree of teacher assistance offered each group differed in terms of time and content. Suggestions and feedback from the teacher were seen as desirable and essential at some points during the process but perhaps not throughout the experience.

Moreover their enjoyment of the task was obvious to the researcher. Any nursing educator who has attempted to teach nursing care planning would realize the importance of this finding. The students could immediately see the application of this care plan in relation to the patients they were caring for in the clinical area. The interaction and immediacy of application to care allowed them to view the planning as direct nursing care not just an assignment.

Conclusions

The findings of this exploratory study indicate that students' understanding of nursing care planning can be enhanced and new insights gained by the use of the computer conferencing system, CO-CO. Since no other micro-computer-based conferencing system exists for real-time, synchronous interaction like CO-CO, the uses for such a system need to be explored and tested.

The limitations of such a study are evident. The enhancement of learning was not measured by any means other than in the students' perception. Since the questionnaires were not anonymous, the students may have experienced some pressure to respond positively to the learning situation. Moreover, since it was a new learning situation, the students may have reacted more positively to the experience. If it became more routine the uniqueness of the learning situation might be lessened.

The preliminary data suggests that CO-CO is a medium worth exploring. As the literature points out computers have not yet been used to their potential in nursing education. Instead we have focused on its more obvious uses to research, word processing and single-user programmed software dealing with specific content.

The other important aspect of the experience was the self-reported improvement of their group process. Students in a nursing program must learn to work effectively within a group and to maximize the group's resources. Their comments concerning the use of CO-CO to interact more effectively were an added feature to the nursing care planning.

Perhaps with the use of a computer conferencing tool, such as CO-CO, nursing educators can allow the students to more creatively deal with problem situations such as care planning. In so doing, the students may also learn to more effectively work as a group in an independent manner while allowing the nursing educator a measure of freedom to engage in other other work-related activities. The use of this type of conferencing system within nursing curricula must continue to be explored. The recent announcement by the developing company, Groupware Technologies Corp., that a modem-based version of the software called TELE-CO-CO is now being produced with funding assistance from the National Research Council of Canada significantly broadens the potential uses and scope of this new and exciting software.

REFERENCES

Beckwith, Don. (1987). Group problem-solving via computer conferencing: The realizable potential. Canadian Journal of Educational Communication. 16(2), 89-106.

Bitzer, M. and Bitzer, D.L. (1973). Teaching nursing by computer: and evaluative study. Nursing Outlook. 32 (1), 34-36.

Blowmire, Robert. (1988). Visiting professor. University of Houston. In Conversation, Althouse College, The University of Western Ontario.

Hassett, Mary. (1984). Computers and nursing. Education in the 1980's. Nursing Outlook. 32 (1), 34-36.

McCalla, Gordon I. (1987). The promise of artificial intelligence for computer assisted learning. Proceedings of the International Conference on Computer Assisted Learning in Post-Secondary Education. (p.31-36). Calgary, Alberta, Canada: The University of Calgary.

Meadows, Lynda S. (1977). Nursing education in crisis: a computer alternative. Journal of Nursing Education. 16 (5), 13-21.

Saba, V.D. & McCormick, A. (1986). Essentials of Computers for Nurses. Philadelphia: Lippincott.

Walker, Mary Boose. (1986). Nursing Education. Challenges of the Computerized Environment. Computers in Nursing. 4 (4), 166-171.

Wilton, Jo Ann. (1987). Evaluation of the Software Package, CO-CO. Unpublished manuscript.

HYPERMEDIA/HYPERMÉDIA

ANIMATION IN SCIENCE TEACHING: TECHNIQUES AND EXAMPLES

R. Ransom
Faculty of Science
The Open University
Milton Keynes, UK

Static pictures in science textbooks are a poor way of representing dynamic processes and the animation of processes is an excellent way of allowing students to comprehend 'difficult' subject areas.

Computer animation has traditionally been a high cost, labour intensive area, but the introduction over the past four years of personal computers with superior graphical interfaces and fast processors has lead to the appearance of graphics and animation tools within the reach of the university or college teacher. In particular, the Apple Macintosh and Commodore Amiga computers benefit from several advnced animation systems (VideoWorks, DeLuxe Video, Aegis Animator) that offer rapid and quick methods for developing visually dynamic teaching systems.

This talk will describe how animations can be built up using these various packages, describing common features and differences between them. Examples will be drawn from the area of the Biological Sciences, in particular genetics and cell biology. Of particular interest is the way in which animations can be spliced into Macintosh HyperCard teaching stacks.

The Open University has a strong involvement in developing efficient computer-aided instruction methodologies, and these animation systems are being developed for use in our Summer School and Day School programmes.

L'ANIMATION DANS L'ENSEIGNEMENT DES SCIENCES: TECHNIQUES ET EXEMPLES

L'utilisation d'images fixes dans les livres d'enseignement des sciences constitue une façon bien malhabile de représenter des processus dyamiques tandis que l'utilisation d'images animées facilite grandement aux élèves la compréhension de questions "difficiles".

L'animation par ordinateur a toujours été un secteur d'activité à coût élevé et utilisant beaucoup de main-d'oeuvre; cependant, au cours des quatre dernières années, l'introduction d'ordinateurs personnels dotés d'interfaces graphiques évolués et de processeurs rapides a permis de mettre des outils graphiques et d'animation à la portée des enseignants de niveau collégial ou universitaire. De façon plus particulière, le Macintosh de Apple et le Amiga de Commodore ont été munis de plusieurs systèmes évolués d'animation (VideoWorks, DeLuxe Video, Aegis Animator) permettant de mettre au point rapidement des systèmes d'enseignement visuels dynamiques.

L'exposé présente les modalités de création de dessins animés au moyen de ces divers progiciels ainsi que les similitudes et les différences entre chacun, avec des exemples dans le domaine de la biologie, surtout en génétique et en biologie cellulaire. La possibilité de dissémination des dessins animés dans divers didacticiels HyperCard Macintosh présente un intérêt particulier.

La Open University prend une part active à l'élaboration de méthodes efficaces d'enseignement assistées par ordinateur et ces systèmes d'animation sont actuellement mis au point pour utilisation dans le cadre de nos programmes d'école d'été et d'école de jour.

ANIMATION IN SCIENCE TEACHING: TECHNIQUES AND EXAMPLES

R. Ransom
Faculty of Science
The Open University
Milton Keynes, UK

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

HYPERCARD ET LES FONCTIONNALITÉS DE L'EAO

B. de La Passardière
Laboratoire MASI
Université Pierre et Marie Curie (Paris VI)
Paris, France

Dans cette communication, nous étudions le logiciel de développement HyperCard (Macintosh, Apple) pour savoir dans quelles mesures il satisfait les fonctionnalités spécifiques de l'Enseignement Assisté par Ordinateur. Nous examinons les facilités offertes et les comparons aux fonctions habituellement présentes dans d'autres systèmes-auteurs.

A cet effet, nous avons créé sous HyperCard un didacticiel qui a déjà fait objet de réalisation sous d'autres logiciels spécialisés en l'EAO. Par ce travail, nous souhaitions aller au délà de la fonction de navigation très particulière d'HyperCard et des possibilités visuelles du Macintosh pour nous intéresser davantage aux possibilités d'interactivité de l'élève en phase d'apprentissage.

Pour pallier le manque de certaines fonctionnalités (analyse de réponses, suivi automatique ..) et pour faciliter le développement de logiciels éducatifs avec HyperCard, nous avons défini un environnement de programmation spécifique. Notre exposé sera étayé de nombreux exemples de réalisation et présentera des cartes-prototypes: canevas pour réaliser certains exercices.

HYPERCARD AND THE FUNCTIONS OF CAL

In this paper, we study the use of the HyperCard development tool (Macintosh, Apple) as a CAL system. We examine the facilities provided and compare these with the functions usually found in authoring languages and systems.

For this study, we created a dedicated version of a special courseware, which had already been previously developed under other CAL Systems. In this work we wanted to go further than simply the use of the browse tool of HyperCard and the Macintosh graphics, as we are more interested in interactivity for the learning student.

Since many CAL functions are missing, we have designed a programming environment to enable the development of educational stackware. In the presentation, we will give a number of examples and present prototype cards.

HYPERCARD ET LES FONCTIONNALITES DE L'EAO

B. de LA PASSARDIERE

Université Pierre et Marie CURIE (PARIS VI)
Laboratoire MASI

Depuis quelques années, l'équipe à laquelle j'appartiens s'intéresse aux logiciels d'aide à la création de didacticiels sur micro-ordinateur. Elle a notamment publié une étude comparative [1] et une série de rapports sur les langages et systèmes-auteur étudiés. Dans le cadre de cette recherche, nous avons conçu un scénario spécifique pour tester les différentes fonctionnalités de l'Enseignement Assisté par Ordinateur (EAO). Sur la base de ce scénario, des didacticiels ont ensuite été réalisés en utilisant plusieurs systèmes-auteur. Bien que **HyperCard**[1] ne soit pas a priori un logiciel d'EAO, il nous a paru intéressant de l'inclure dans l'étude. Dans cet article, HyperCard n'est étudié que du strict point de vue de l'EAO. En effet, nous n'y envisageons pas toutes les possibilités du logiciel, mais nous nous restreignons aux seules fonctionnalités habituellement offertes dans les systèmes d'EAO tant pour l'auteur que pour l'élève.

PRESENTATION

HyperCard est un outil de développement sur Macintosh[1]. Il bénéficie donc de l'environnement particulièrement convivial de cet ordinateur (souris, interface graphique, communication entre applications...) qui en fait apparemment un logiciel assez facile d'utilisation et un média bien adapté à l'Enseignement Assité par Ordinateur. HyperCard peut être vu comme une base d'informations associée à un langage orienté objet. Par bien des aspects, il est la manisfestation de concepts issus de SmallTalk avec la notion d'objets et la programmation par messages. Avant d'étudier HyperCard en tant qu'outil de création de logiciels éducatifs, nous rappelons les grandes lignes de son fonctionnement et nous fixons la terminologie utilisée.

Un *OBJET* est une entité caractérisée par des propriétés statiques et par des propriétés dynamiques. Les premières précisent les caractéristiques de l'objet et spécifient la manière dont l'objet se présente : forme, apparence, nom... Les secondes décrivent les actions exécutées lors de l'activation de l'objet. Pour situer ces deux caractéristiques dans le cadre particulier de l'EAO, on peut considérer que les propriétés statiques précisent comment l'objet est vu par l'élève tandis que les propriétés dynamiques décrivent les réactions du didacticiel aux interventions de l'élève (réponse, demande d'aide, dictionnaire, arrêt...).

On distingue plusieurs types d'objet : les piles, les cartes, les fonds de carte, les boutons et les champs. Une *PILE* est un ensemble ordonné de *CARTES* dont on sait distinguer notamment la première, la dernière ou encore la suivante. Une carte est le résultat de la superposition d'un *FOND* et d'informations : textes, gaphiques, dessins, sons ou autres. Les *BOUTONS* sont des objets particuliers qui réagissent aux actions de la souris. Les *CHAMPS* sont, quant à eux, des zones délimitées recevant exclusivement du texte. Boutons et champs peuvent figurer aussi bien dans la carte que dans le fond.

A chacun de ces objets est associé un *SCRIPT* écrit en **HyperTalk**[1], le langage de programmation de HyperCard. Un script est composé d'un ensemble de `routines`(*gestionnaires de messages*) dont l'en-tête spécifie le *message* ou événement auquel l'objet va réagir et dont le corps précise comment il va y réagir et notamment quels sont les messages qu'il va à son tour émettre vers d'autres objets. Un message est pris en charge par le premier objet dans la structure hiérarchique capable de le traiter. En dernier recours, le message est traité par HyperCard lui-même (cas des commandes du langage).

Le langage HyperTalk [2, 3] est un langage de programmation. Comme dans ceux-ci, on trouve des notions voisines de procédures et de fonctions (`on`, `function`), différentes formes de structures répétitives (`repeat for`, `repeat until`, `repeat while`, `repeat forever`, `repeat with`), différentes formes de structures conditionnelles (`if ... then ... else ... endif`) et la notion de variables (locales, globales ou prédéfinies). A cette panoplie de structures usuelles, il faut ajouter un ensemble assez complet de commandes dont, par exemple, des commandes spécifiques aux objets (`get`, `hide`, `show`, `set`) ou à la navigation(`find`, `go`, `push`, `pop`, `show`) ou encore des fonctions particulières comme les fonctions temps (`date`, `secondes`, `ticks`, `time`). On dispose aussi dans le langage de toutes les possibilités graphiques du *menu Outils* (`choose`) ainsi que d'opérateurs logiques

1 Macintosh, HyperCard, HyperTalk sont des marques déposées d'Apple Computer Inc.

(`contains, is in, is not in`) et de commandes sur les propriétés des objets (`autoHilite, lockScreen, visible`). En HyperCard, les objets dialoguent par messages (certains envoyés par le système tels que `mouseUp` ou `openCard`) et la programmation est répartie sur les différents objets. L'auteur doit donc choisir le meilleur objet pour placer le script et c'est sans doute là une des difficultés d'apprentissage d'HyperTalk.

FONCTIONNALITES POUR L'EAO

Dans ce paragraphe, nous passons en revue les principales fonctionnalités des systèmes d'EAO et leur réalisation en HyperCard.

* *le choix du niveau*

Par rapport au découpage classique en EAO de Mode Auteur et Mode Elève, on trouve une palette beaucoup plus large. En effet, il existe en HyperCard plusieurs niveaux de puissance ou d'utilisation :

Niveau	Nom		Utilisation
1	Navigation	(broswing)	explorer
2	Texte	(typing)	écrire
3	Dessin	(painting)	dessiner
4	Auteur	(authoring)	modifier
5	Programmation	(scripting)	créer

Le **mode AUTEUR** (au sens EAO) correspond au *niveau 5* (programmation) et non au niveau 4 comme le nom pourrait le laisser croire. En effet, l'auteur d'un didacticiel doit non seulement prévoir le graphisme des cartes mais aussi les scripts, c'est à dire des actions associées à chacun des objets ainsi que les liens qui unissent les différentes cartes. Le *niveau 4* correspond davantage au cas de l'enseignant qui, disposant d'un logiciel éducatif, souhaite y apporter quelques changements mineurs pour l'adapter à ses propres besoins. Ce niveau autorise l'accés au texte, aux outils de peinture, aux boutons et aux champs, mais pas aux scripts ; c'est-à-dire que l'utilisateur peut changer l'apparence des objets mais pas les actions associées. Le *niveau 2* correspond exactement au **mode ELEVE** dans la mesure où il permet la saisie de réponses textuelles en plus des réponses par désignation autorisées au niveau 1. De plus, ce niveau offre l'avantage de n'afficher qu'une barre de menu réduite. Dans certains cas comme celui d'un logiciel de découverte (exploration d'un ensemble des connaissances mis à la disposition de l'élève), on peut resteindre le mode élève au seul niveau 1. A l'inverse, on peut aussi souhaiter offrir à l'élève des possibilités graphiques et donc étendre le mode élève au niveau 3. Mais dans ce cas-là, l'étendue des possibilités offertes rend difficile l'analyse de réponses. De plus, il faut se prémunir contre les destructions ou altérations malencontreuses, car HyperCard enregistre immédiatement tout changement opéré. Ainsi ce que l'élève gagne en possibilités, l'auteur le perd en contrôle.

```
MODE AUTEUR = niveau 5 (programmation)
MODE ELEVE = niveau 2 (texte)
```

* *le choix de la structure du didacticiel*

Dans le cas d'HyperCard, un didacticiel n'est autre qu'une pile. Néanmoins, il peut être souhaitable de le subdiviser en plusieurs piles, liées les unes aux autres. De même, si le didacticiel comprend des parties autonomes avec des présentations très différentes du reste, on peut imaginer le didacticiel comme un ensemble de piles, chacune d'elles ayant sa spécificité.

Comme pour tout autre système-auteur, l'auteur-concepteur doit, dans une première phase, découper le contenu pédagogique de son projet en un ensemble d'entités logiques ou d'unités de traitement élémentaires. En général, chacune est décrite par une carte. Néanmoins une situation complexe peut se traduire soit par une succession de plusieurs cartes visuellement assez proches les unes des autres (animation, aide...), soit par la superposition de nouvelles informations sur la carte initiale (renforcement, commentaire, complément, aide...). L'auteur doit ensuite déterminer si le découpage en unités élémentaires donne lieu à la création d'une seule ou plusieurs piles et d'un ou plusieurs fonds de carte. Néanmoins, cette analyse qui constitue le principal travail de tout concepteur de didacticiel n'est pas le propre d'HyperCard.

```
En général, un didacticiel = une pile
Parfois, un didacticiel = plusieurs piles
```

* *les facilités de gestion de didacticiels*

Comme d'autres langages et systèmes-auteur, HyperCard posséde des facilités de gestion de didacticiels : créer, modifier, dupliquer, renommer, effacer. Il est par contre difficile de faire la fusion de deux didacticiels. En effet, importer dans une pile une carte provenant d'une autre pile revient à créer un autre fond ce qui est parfois souhaitable pour engendrer un deuxième fond, mais le plus souvent indésirable.

```
Fusion de 2 didacticiels = difficile
```

* *le fond de carte*

Le fond de carte regroupe les informations communes à toutes les cartes ou celles qui sont fréquemment présentes. Cette notion de fond, que l'on trouve aussi dans des langages-auteur comme ARLEQUIN permet de créer des didacticiels homogènes tant du point de vue de l'ergonomie que des requêtes disponibles. Pour **créer un fond** de carte, l'auteur peut soit partir d'un fond existant (dans la pile STACK IDEAS ou toute autre pile du domaine public) et l'adapter à ses besoins, soit définir son propre graphisme (*Nouveau fond*).

• *le contenu*

Les connaissances à transmettre, qu'elles soient textuelles ou graphiques sont surtout contenues dans la carte. Pour **créer une carte**, l'auteur peut, comme pour le fond, soit partir d'une carte existante, soit demander une carte vierge ne comportant que les éléments du fond courant (*Nouvelle carte*).

le texte

Un texte est soit un **texte peint** soit un **champ**. Ces deux types de texte sont traités de manière fort différente. En effet le texte est vu, dans le premier cas, comme une partie d'un dessin que l'on peut pivoter, inverser et dans le second cas, comme une zone de texte que l'on peut éditer avec toutes les fonctions usuelles d'édition de texte : copier, coller, effacer... Dans les deux cas, on peut en définir le style, l'alignement, la police, la taille des caractères et l'interlignage. En raison des facilités d'édition et donc de mise au point, on utilisera davantage les champs.

le fenêtrage

La fenêtre créée par l'auteur n'existe pas en tant que telle en HyperCard. L'objet qui s'apparente le plus à cette notion est le champ qui, de fait, définit une zone. Mais cette zone est exclusivement textuelle. De plus, le texte qui est affiché dans cette zone n'est pas justifié automatiquement à droite et à gauche (il l'est uniquement à gauche). L'auteur doit donc pour cadrer correctement son texte dans la fenêtre gérer lui-même les espacements entre les mots. S'il le désire, l'auteur peut verrouiller le champ, empêchant ainsi toute modification accidentelle du texte. Dans ce cas là, le curseur de navigation ne change pas de forme en passant au dessus du champ. L'auteur n'a pas la possibilité de définir une zone pour l'affichage d'un dessin.

le graphique

HyperCard dispose sans conteste d'un jeu très complet d'outils pour composer des cartes au graphique soigné. Dans HyperCard, on retrouve toutes les performances des logiciels tels que SuperPaint ou MacDraw, auxquelles viennnent s'ajouter de nouvelles facilités obtenues par combinaison des outils avec les touches *Option* ou *Commande* et le *double-clic* : menu détachable, forme remplie avec ou sans bord, effacement de tout le dessin, défilement de l'image grossie... En plus de ses performances intrinsèques, HyperCard autorise l'importation de graphismes ou dessins créés par d'autres applications via le presse-papier ou l'album notamment.

```
contenu des cartes = outils multiples
```

• *les exercices*[2]

Comme HyperCard n'est pas un système spécialisé pour l'enseignement, il n'existe aucun type d'exercice prédéfini. Néanmoins la richesse du langage HyperTalk permet de combler cette absence. En effet, quelles que soient la forme de l'exercice proposé et celle de la réponse attendue, il est possible d'en faire l'analyse par le script d'un objet. Le mode de saisie priviligié dans l'environnement Macintosh est bien sûr la souris, mais il n'exclut nullement la saisie au clavier. La réponse est alors saisie dans un champ. Dans le cas de réponses multiples, l'élève tape sa réponse dans différents champs en utilisant la touche *Tabulation* pour se déplacer de champ en champ. Un champ n'est pas limité à son contour graphique, autrement dit même si la zone affichée semble ne pouvoir contenir que quelques caractères, la taille du champ est en fait quasi illimitée (32 767 caractères). En conséquence, toute vérification sur la longueur ne peut être réalisée que par programmation. Il en va de même si l'on souhaite n'autoriser que certaines touches du clavier. De ce fait, lorsque la réponse ne doit comporter qu'un caractère ou un alphabet restreint, on choisira plutôt une réponse au moyen de la souris que du clavier.

Pour réaliser les questions à choix multiples, on peut utiliser un bouton par choix tandis que, pour les questions à trous on peut utiliser des champs. Dans bien des exercices, il faut ajouter un bouton "Fin" pour déclencher l'analyse de réponses. En effet, si on autorise l'élève à modifier ses réponses, il faut pouvoir repérer lorsqu'il a terminé.

Les exercices en HyperCard peuvent prendre beaucoup de formes, dont seule la complexité de réalisation freinera l'imagination.

```
exercices = à programmer
```

• *l'analyse de réponses*

Il n'existe pas davantage d'analyse de réponses. Il faut donc pour pallier cette absence introduire des fonctions écrites en HyperTalk. Par exemple, les fonctions :

présent c,t	– la chaine c présente dans le texte t ?
absent c,t	– la chaine c absente du texte t ?
sprésent l,t	– un mot dela liste l présent dans t ?
majuscule t	– mettre le texte t en majuscule

Le résultat de ces fonctions peut être combiné par les opérateurs booléens habituels : `and`, `or`, `not` ou par d'autres opérateurs créés, eux aussi, par programmation. Il n'y a pas non plus de "jokers", c'est à dire de caractères particuliers pour préciser des tolérances orthographiques. Pour gérer les synonymes, on doit là encore avoir recours à des fonctions écrites en HyperTalk comme la fonction *sprésent* dans laquelle les synonymes sont donnés sous forme d'une liste d'éléments.

Pour les réponses numériques, on est confronté au même problème d'analyse de réponses. Hormis les classiques fonctions arithmétiques (`abs`, `average`, `min`, `max`, `sin`, `cos...`) et les commandes arithmétiques (`add`, `divide`, `substract`, `multiply`), il n'existe pas de fonctions spécifiques pour accepter une réponse si elle est comprise dans un intervalle, déterminer la précision du résultat, établir des

2 Faute de place, les scripts du didacticiel réalisé n'ont pu être joints à ce texte ; ils sont données dans [5].

synomymes entre les unités de mesure... L'auteur doit donc prévoir et écrire tous ces utilitaires.

```
analyse de réponses = à programmer
```

* *la notation*

Il n'y a, bien sûr, pas de notation intégrée dans HyperCard. Néanmoins, il est assez facile de l'intégrer à différents niveaux dans les scripts. On peut initialiser une variable globale dans le script de la pile (`on openStack`), puis mettre à jour cette variable localement.

```
notation = à programmer
```

* *le cheminement*

La liaison entre les différentes unités élémentaires est réalisée dans le script du bouton qui permet de passer à la suite (option `"Lier à"` ou `go to`).

```
cheminement = intégré
```

* *le suivi*

Le logiciel HyperCard étant un outil général de développement, il n'existe pas plus de suivi élève que de notation, ces deux notions étant très liées à l'Enseignement Assisté par Ordinateur. Là encore, HyperCard permet de résoudre le problème de différentes manières en créant soit un fichier, soit une autre pile, soit un champ spécial dans la pile.
Pour la gestion du suivi (collection des différents suivis, édition, statistiques, visualisation graphique...), il faut une fois de plus écrire des scripts en HyperTalk. Mais on peut aussi utiliser d'autres applications comme Multiplan ou Cricket Graph mieux adaptées à ce genre de traitement ; pour cela, il suffit d'introduire dans le fichier créé pour le suivi des caractères de *Tabulation* et de *Retour Chariot* reconnaissables par ces applications.

```
suivi = à programmer
```

* *l'aide*

L'élève peut bénéficier d'une aide globale ou locale, dès qu'un bouton d'aide est mise à sa disposition. Pour réaliser ce double niveau, on peut définir un bouton d'aide dans le fond de la carte pour l'aide globale, puis couvrir ce bouton par un bouton identique au niveau de la carte pour l'aide locale. En effet, HyperCard explorent les objets selon un ordre hiérarchique, il traite en premier le bouton de la carte (aide locale) s'il existe et le bouton de fond (aide globale) sinon. On peut aussi bien entendu créer deux boutons différents.

* *la mise au point*

Contrairement à beaucoup de systèmes-auteur qui ont un mode auteur et un mode élève séparé, HyperCard est très pratique pour la mise au point. En effet, il est possible d'exécuter immédiatement un script nouvellement créé ou modifié en sélectionnant l'outil de Navigation. Le passage d'un mode à un autre est d'autant plus aisé qu'il est possible de détacher le menu Outil de la barre des menus.

CONCLUSION

On pourrait encore longuement considérer les possibilités d'HyperCard. Néanmoins, cette étude permet déjà de tirer quelques conclusions.
HyperCard est un outil de développement général pouvant être utilisé dans de nombreux domaines. En Enseignement Assisté par Ordinateur, les boutons et la commande de navigation à travers une pile en font un outil très séduisant pour réaliser des didacticiels d'exploration (démonstration, parcours arborescent, dictionnaire visuel). Cependant un logiciel éducatif ne se contente généralement pas de parcourir un ensemble de connaissances, il doit aussi être interactif. De ce point de vue, il manque des fonctionnalités aussi importantes que l'analyse de réponses. Néanmoins, à l'aide du langage HyperTalk, il est possible de réaliser quasiment n'importe quelle fonction et donc de créer un environnement adéquat.
En l'état actuel, de nombreux auteurs doivent se décourager car, pour arriver à réaliser ce qu'ils souhaitent, ils doivent maîtriser HyperTalk. Or la richesse de ce langage est telle que son apprentissage reste fort long pour un auteur désirant réaliser rapidement un didacticiel.
En conséquence, HyperCard est un logiciel de développement qui permet de créer facilement des logiciels d'exploration dans lesquels le travail se limite presque uniquement à la création des cartes graphiques. Il en va tout autrement pour réaliser des logiciels éducatifs avec des réponses textuelles. C'est pourquoi nous avons développé un environnement adapté à l'EAO [4]. Cet environnement a été testé dans le cadre d'un projet européen de didacticiels en génie logiciel.

REFERENCES

[1] F. MADAULE, P. BARRIL, B. de LA PASSARDIERE, F. LE CALVEZ, M.M POC, M. URTASUN. "Systèmes d'Enseignement Assisté par Ordinateur". Revue T.S.I, vol 6, n° 1 , 1987

[2] Frédéric RINALDI. "HyperCard : la programmation en HyperTalk". Cedic Nathan, 1988

[3] Danny GOOGMAN. "The HyperCard handbook". Bantam Books,1988

[4] Brigitte de LA PASSARDIERE. "A CAL Environment for HyperCard". ICAL89, Dallas,1989

[5] Brigitte de LA PASSARDIERE. "Quels outils informatiques pour réaliser des didacticiels ? une étude sur HyperCard". Université ParisVI, Rapport MASI n° 250,1988

HYPERMÉDIA: UNE TECHNOLOGIE ACCESSIBLE POUR LE PROTOTYPAGE EN EIAO: LE SYSTÈME E C O

D. Déry, J. Imbeau, P. Duchastel
Lab. d'intelligence artificielle en éducation
Université Laval
Québec, PQ

Hypermédia constitue une véritable boîte à outils pour emmagasiner, gérer et retrouver de l'information. Elle dépasse les logiciels conventionnels car nous n'avons plus à suivre un cadre d'organisation prédéterminé pour y mettre et y retrouver de l'information.

Depuis que cette technologie a fait son apparition sur le marché, les concepteurs de la technologie pédagogique peuvent bénéficier de son accessibilité et de la rapidité avec laquelle elle facilite la conception et la production. En effet, Hypermédia est facile à manipuler et le concepteur peut en quelques heures obtenir un prototype du système désiré.

Nous désirons illustrer ici, les possibilités de cette technologie que nous avons explorées dans notre recherche pour la conception d'un prototype sur l'écologie des populations (ECO). Le système ECO est en développement continu à l'Université Laval. Ce prototype propose une application d'HyperCard sur appareil Macintosh.

HYPERMEDIA: ACCESSIBLE TECHNOLOGY FOR PROTOTYPING ICAL: THE E C O SYSTEM

Hypermedia is really a toolbox of items for storing, controlling and retrieving information. It goes beyond conventional software, since we no longer have to follow a predetermined organizational framework to store and retrieve information.

Since this technology came onto the market, designers of instructional technology have benefited from its accessibility and the speed with which it facilitates design and production. Indeed, Hypermedia is easy to manipulate and the designer can obtain a prototype of the desired system in a few hours.

Our purpose in this paper is to illustrate the possibilities we have explored for this technology in our research into the design of a prototype on the ecology of populations (ECO). Development of the ECO system continues at Laval University. This prototype proposes an application of HyperCard on Macintosh equipment.

HYPERMEDIA: UNE TECHNOLOGIE ACCESSIBLE POUR LE PROTOTYPAGE EN EIAO: LE SYSTEME E C O

D. Déry, J. Imbeau, P. Duchastel

Laboratoire d'intelligence artificielle en Education
Université Laval, Québec

RESUME

La technologie Hypermédia telle qu'offerte dans le logiciel HyperCard et certaines techniques issues de l'intelligence artificielle ont constitué nos outils pour la conception d'un prototype de système d'enseignement (E C O).

Le système E C O est un logiciel de simulation sur l'écologie des populations. Il veut favoriser l'acquisition de la démarche scientifique et des connaissances en écologie. Ce prototype est en développement sur un appareil Macintosh SE.

INTRODUCTION

Depuis que la technologie Hypermédia a fait son apparition sur le marché, les concepteurs de la technologie pédagogique peuvent bénéficier de son accessibilité et de l'accélération du rythme de conception et de production qu'elle permet. En effet, l'interface Hypermédia est facile à manipuler et le concepteur peut en quelques heures obtenir un prototype du système désiré.

Nous désirons illustrer ici, les possibilités de cette technologie que nous avons explorées lors de la conception d'un prototype de système d'enseignement. Le système E C O simule l'écologie des populations, dans le but d'aider l'étudiant de niveau secondaire à appliquer la démarche scientifique tout en assimilant des notions d'écologie.

DESCRIPTION GENERALE

La plupart des auteurs (v.g. Anderson, 85; Yazdani, 87) s'entendent maintenant sur les caractéristiques générales que doit posséder un système tutoriel intelligent soit:

- offrir un dialogue interactif riche et accessible,
- connaître le domaine,
- connaître l'étudiant et
- savoir enseigner.

Nous avons élaboré le système E C O selon ce modèle. Il possède les quatre composantes correspondantes:

- une interface de communication,
- un module expert,
- un modèle étudiant et
- un module tutoriel.

Les 4 modules sont initiés mais ils ne sont pas complétés, ni entièrement fonctionnels. L'architecture du système est ouverte: les 4 modules sont distincts. Le format simulation retenu nous permet de modéliser de façon simple un écosystème complexe et d'isoler facilement les variables à étudier. Il offre à l'étudiant un environnement qui l'invite à concevoir, explorer, prédire et vérifier des faits concrets, sans l'obliger à suivre un cheminement prédéterminé (Yazdani, 88).

UNE DEMARCHE SCIENTIFIQUE FAVORISANT L'APPRENTISSAGE

Dans ce projet, nous nous inspirons de l'approche de Papert (80) selon laquelle l'ordinateur devrait constituer un environnement où l'apprenant devrait pouvoir prendre l'initiative. Avec Paquette (88) et Bergeron (88), nous privilégions "l'utilisation de l'ordinateur comme un outil pour apprendre plutôt que comme une machine à enseigner".

En situation d'apprentissage, tout processus d'acquisition des connaissances implique un traitement adéquat de l'information de la part de l'apprenant. Nous pouvons améliorer ce processus en fournissant des environnements constructifs et motivants supportant efficacement la démarche scientifique. Spada & al. (83) ont proposé une démarche scientifique satisfaisante pour nos besoins et notre clientèle, la séquence "S-P-F-P" pour "Search-Predict-Feedback-Process", i.e.

- chercher l'information: faire des observations;
- faire des prédictions: émettre des hypothèses;
- tenir compte des feedbacks: expérimenter, vérifier;
- traiter l'information: analyser, conclure, synthétiser.

Par conséquent, en utilisant cette démarche, nous comptons favoriser les habiletés suivantes chez l'étudiant:

- acquisition de connaissances en écologie,
- application des lois et des principes pertinents,
- compréhension et interprétation des faits, des graphiques,des phénomènes et des relations,
- analyse des implications des hypothèses,
- validation des inférences,
- capacité d'analyse et de synthèse (Powell, 84; Schiff & Kandler, 88).

Cette approche de Spada & al. (83) nous semble bien indiquée ici, autant sur le plan de l'efficacité, que de la facilité et de l'accessibilité. Soulignons qu'elle nous intéresse surtout à cause du processus cognitif de résolution de problème qu'elle implique, notre but étant de donner à l'étudiant l'occasion d'appliquer les notions acquises lors du cours théorique. De cette manière l'apprenant acquiert non seulement des connaissances factuelles du domaine simulé mais aussi des habiletés générales concernant le processus de traitement de l'information (Papert, 80; Bergeron, 88).

FONCTIONNEMENT DU SYSTEME E C O

En utilisant le logiciel, l'étudiant se construit un écosystème réduit où évoluent les espèces qu'il a choisies. Il travaille sur des problèmes dont il définit lui-même les variables. En évoluant dans ce "micromonde" , il peut inventer, manipuler et vérifier certaines hypothèses et selon ses intérêts et ses besoins, il peut remodeler, restructurer ou rebâtir un nouvel écosystème en fonction de ce qu'il veut expérimenter ou vérifier (Papert, 80).

Au début d'une session de travail sur ECO, l'étudiant est invité à entrer les données de l'écosystème à simuler, d'abord

- la ou les espèces en présence,

puis pour chacune:

- la population initiale,
- le taux annuel de natalité,
- le taux annuel de mortalité et
- le taux annuel de prédation .

Ces éléments constituent les variables décrivant la population dont le système ECO simule l'évolution sur une période de 5 ans. L'élève est invité à prédire l'évolution de la population. Il est incité à représenter ses données et calculs sous forme de tableaux et graphiques.

Le système procède selon les étapes de la séquence "S-P-F-P" décrite précédemment. Il questionne l'étudiant et enregistre ses réponses à la fois dans les modules expert et étudiant pour les fins de traitement des données. Le module expert résout le problème soumis quand toutes les données nécessaires sont entrées. Le module tutoriel diagnostiquera l'état de l'apprentissage en analysant la trace enregistrée dans le modèle étudiant et déterminera la meilleure règle tutorielle ou stratégie d'enseignement à appliquer. Tout au long de l'interaction un bouton "INFO" permet à l'étudiant d'obtenir de l'aide selon ses besoins. La session prend fin quand l'étudiant a résolu son problème et répondu aux questions posées par le système. La figure 1 présente un exemple du dialogue initial qui s'établit entre un étudiant et le système ECO. Elle présente également quelques exemples de questions que le système pose par la suite.

ECO: -Quel est ton prénom?
Etudiant: -Denis
ECO: -Quel est ton nom?
Etudiant: -Lajeune
ECO: -As-tu déjà joué à ECO?
Etudiant: -Non
ECO: -Clique sur l'espèce dont tu veux étudier la dynamique de la population
Etudiant: -Aigle
ECO: -Combien d'individus compte la population d'aigles?
Etudiant: -5
ECO: -Quel est le taux de natalité de cette population par an?
Etudiant: -1
ECO: -Quel est le taux de mortalité de cette population par an?
Etudiant: -2
ECO: -Quel est le taux de prédation contre cette espèce par année?
Etudiant: -0
.
.
(ensuite le système pose des questions comme:)
.
.
ECO: -Quel sera le taux de la population dans 2 ans?
.
ECO: -Dans combien d'années cette population va-t-elle s'éteindre?
.
ECO: -Cette population est-elle en croissance? en décroissance?
.

Fig. 1 Bribes du dialogue d'une intéraction avec le système ECO

LES COMPOSANTES DU SYSTEME

Comme mentionné plus haut, le système comprend quatre composantes:

- une interface de communication gérant les dialogues élève-machine; elle offre des boutons à cliquer à l'aide de la souris;

- un module expert comprenant une base de faits, une base de règles et un résolveur de problème. Ce module est interactif; un bouton INFO permet, selon les besoins, de poser des questions au système;

- un modèle étudiant qui garde la trace des actions de l'apprenant lors des interactions avec le système. Il contient un catalogue des fausses conceptions et une trace des réponses de l'étudiant pour décrire son état immédiat d'apprentissage; cette trace est constamment mise à jour et assure le raffinement progressif du modèle.;

- un module tutoriel (non encore progammé) qui comprendra des règles tutorielles incorporant des stratégies d'enseignement et un analyseur de réponses pour comparer les solutions de l'étudiant à celles de l'expert.

L'INTERFACE DE COMMUNICATION

Le système offre des menus réduits à des boutons (Fig. 2) que l'on clique à l'aide de la souris. La communication est simple, rapide et efficace. Pour maximiser la compréhension et la mémorisation de l'utilisateur nous avons limité l'usage des textes, préférant les remplacer par des images, des graphiques ou des tableaux. Cette interface nous permet d'aller chercher, sous une forme suffisamment raffinée, l'information nécessaire pour modéliser l'étudiant. Le mode INFO donne accès à des menus où l'étudiant peut cliquer pour interroger le module expert.

Fig. 2 Interface: Menu de boutons à cliquer

LE MODULE EXPERT

Tel que déjà mentionné, le module expert comprend trois parties. Chacune peut facilement être augmentée manuellement, mais non automatiquement car le système n'acquiert aucune connaissance de façon autonome.

La base de connaissances factuelles

Les connaissances thématiques sur l'écologie sont représentées dans une base, sous forme de fiches indépendantes, chacune correspondant à une fiche du logiciel HyperCard. Cette base de faits regroupe les connaissances déclaratives, telles que les définitions usuelles: écosystème, taux, natalité, mortalité, prédation, etc. et les caractéristiques des animaux présents dans les écosystèmes (Fig. 3). Des descripteurs permettent au système de repérer les caractéristiques

BASE DE FAITS AIGLE
animal, aigle
est,oiseau
bec, rapace, de proie, crochu
natalité, 0, 1, 2
mortalité, 0, 1, 2
prédation, 0
locomotion, vole
activité, diurne

Fig. 3 Base de connaissances factuelles

attribuées aux objets. Par exemple, le descripteur "activité" peut prendre un de ces deux qualificatifs, nocturne ou diurne selon l'espèce qu'il caractérise. Ces descripteurs apparaissent dans les menus activés par le bouton INFO.

La base de règles procédurales

Cette base contient les connaissances procédurales i.e. les interrelations entre les faits, les concepts et les règles. Elles prennent la forme de règles de production de type ("IF...THEN..."). Un exemple d'une telle règle pourrait être:

IF TauxMortalité > TauxNatalité
THEN Population Décroissante.

Le résolveur de problème

Le résolveur accumule les données fournies par l'étudiant dès le début de l'interaction. A chaque interaction correspondent des champs précis pour garder la trace des données et le résolveur effectue les calculs quand il possède la totalité des données demandées à l'étudiant. Un tableau sert à compiler les données et les résultats des calculs effectués .

LE MODELE ETUDIANT

Ce module a pour tâche de modéliser l'état des connaissances de l'étudiant: ce qu'il sait, ce qu'il connaît mal, ce qu'il apprend et ce qu'il n'a pas acquis. Il est mis à jour constamment en enregistrant les réponses, les actions et les décisions lors de la simulation (Fum, 88). Cette composante est l'agent activateur du système, elle axe le système sur l'apprenant plutôt que sur le tuteur (Giardina, 88; Papert, 80).

Initialement, le système a peu de connaissances de l'étudiant. Le modèle s'incrémente, se raffine au fur et à mesure de la cumulation des interactions. C'est une modélisation ADAPTATIVE où les traces du modèle changent, s'adaptent selon le processus d'acquisition des connaissances de l'apprenant (Fum, 88). De là, tout son dynamisme à partir duquel nous pouvons construire une interaction plus adaptée, mieux individualisée et possiblement plus efficace (Ikeda, 88; Fum, 88).

Présentement, nous enregistrons dans ce modèle les données (Fig.4.) relatives:

- au nombre d'essai,
- aux espèces choisies,
- aux fausses conceptions,
- aux concepts vus, acquis ou non et
- aux concepts à voir.

MODELE ETUDIANT 1	N-ESSAI	2
AIGLE	5	Espèce Choisie
RENARD	1	Aigle, Renard,
SOURIS	0	
LIEVRE	0	

Fig. 4 Le Modèle Etudiant

Le modèle étudiant doit être conservé à la fin d'une session de travail car il servira de référence au système pour les interactions subséquentes. Comme nous le verrons plus loin ce modèle étudiant est directement relié aux interventions du module tutoriel.

LE MODULE TUTORIEL

Ce module est en développement présentement et il devrait comprendre:

- un analyseur de solution pour évaluer la réponse de l'étudiant en comparaison avec celle de l'expert; au préalable, il procède à l'analyse des fausses conceptions de l'étudiant;

- des règles tutorielles (niveau 1) et des méta-règles (niveau 2);

- des stratégies d'enseignement telles que poser une question, soumettre un problème semblable, y joindre la solution, inciter l'étudiant à utiliser l'INFO, suggérer la simulation d'un cas particulier, faire recommencer une simulation non réussie.

Plusieurs tâches incombent à ce module, il doit entre autres: diagnostiquer, corriger et orienter l'étudiant. Dans notre optique, le système a été conçu de façon à respecter les initiatives de l'apprenant cela de manière à favoriser l'exploration, comme dans les micromondes utilisés par Papert (80). Naturellement, cette approche demande plus d'effort de la part de l'étudiant mais elle se révèle plus intéressante et plus motivante.

Le tutoriel intervient en étroite collaboration avec le modèle étudiant, car ces paramètres sont rattachés à la trace qui y est enregistrée. La compilation de ces paramètres implique des analyses immédiates, épisodiques (Weber, 88) ou globales de l'état des connaissances de l'étudiant. Ces analyses conditionnent les interventions du tutoriel en termes de stratégies d'enseignement, de règles ou de méta-règles. Chaque action de l'étudiant ne déclenche pas nécessairement une réaction du tutoriel, mais elle laisse cependant une trace dans le modèle étudiant, qui pourra être utilisée ultérieurement.

CONCLUSION

Dans cette communication nous avons présenté le système d'enseignement E C O , dont le développement du prototype a été facilité par l'utilisation de la technologie Hypermédia sur Macintosh-SE.

Le système E C O simule l'écologie des populations en suivant les étapes de la démarche scientifique "S-P-F-P" (Spada, 83). Cette approche favorise l'individualisation de l'enseignement ainsi que l'apprentissage. Soulignons que, comme dans les micromondes, cela demande plus d'efforts de chacun mais s'avère des plus enrichissant et des plus motivant (Papert, 80).

Les orientations futures du projet sont le perfectionnement du modèle étudiant et la complexification de la simulation de l'écosystème.

REMERCIEMENTS

Nous tenons à remercier M. P. Foisy et M. B. Laberge, du Département de technologie de l'enseignement, pour leur disponibilité et leur aide concernant la programmation et la présentation visuelle du projet.

REFERENCES

1. Anderson, J. R., Boyle, C.F. et Reiser, B.J.
Intelligent Tutoring Systems. *Science*, 228, 456-462, 1985.

2. Bergeron, A.
Environnements d'exploration, représentation et traitement de l'information. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

3. Bonnet, A.
L'intelligence artificielle Promesses et réalités. InterEditions, Paris, 1984.

4. Cyr, M. et Forget, R.
Cahier d'activités en écologie, Cahier du maître. Editions du Renouveau Pédagogique, Inc., Montréal, 19?.

5. Duchastel, P.
L'intelligence artificielle au service de l'éducation : ALMONDE. Communication présentée au *Colloque Francophone International sur l'Ordinateur et l'Enfant d'âge préscolaire*, Québec, 1987.

6. Furn, D., Giangrandi, P. et Tasso, C.
ET : an Intelligent Tutor for foreign language teaching. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

7. Giardina, M., Marton, P., Duchastel, P. et Doublait, S.
BIOMEC : modélisation et développement d'un module d'apprentissage intelligent utilisant le vidéodisque interactif. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

8. Ikeda, M., Mizoguchi, R. et Kakusho, O.
Design of a General Framework for Intelligent Tutoring Systems. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

9. Le Page, L. et Sauvé-Desrochers, L.
La Planète Bleue, Ecologie. Chevrier, J.-M. (Ed.), Montréal, 1983.

10. Papert, S.
Mindstorms, Children, Computers and Powerful Ideas. Harvester Press/ Basic Books, 1980.

11. Paquette,G.
Le développement d'outils intelligents d'apprentissage pour le traitement des connaissances. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

12. Powell, R.
Letts study aids: A-Level Economics Course Companion. Charles Letts & Co, London, 1984.

13. Ramade, F.
Eléments d'écologie, écologie fondamentale. Mc Graw Hill (Ed.),Paris, 1984.

14. Spada, H., Opwis, K., Donnen, J., Schwiersch, M. et Ernst, A.
Ecological knowledge : acquisition and use in problem solving and in decision making. De Corte, E. (Ed.), *International Journal of Education, 11* , 665-685, 1983.

15. Wenger, E.
Artificial Intelligence and Tutoring Systems, Morgan Kaufman Publishers, inc., Los Altos, CA, 1987.

16. Yazdani, M.
Language Tutoring with Prolog. Communication présentée à la Conférence "*Intelligent Tutoring Systems*", Montréal, 1988.

17. Yazdani, M.
Intelligent Tutoring Systems : An Overview. *Expert Systems, 3* , 154-162, 1987.

ENVIRONNEMENT POUR L'APPRENTISSAGE PROCÉDURAL SUR HYPERCARD

A. Dufresne
Dép. des sciences de la communication
Université de Montréal
Montréal, PQ

La recherche s'inscrit dans la problèmatique de l'enseignement intelligent d'habiletés liées à l'utilisation de l'ordinateur, en particulier pour les progiciels complexes. Elle propose un modèle d'interface pour l'apprentissage du chiffrier électronique, en utilisant le progiciel HYPERCARD. Ce progiciel facilite le prototypage d'interface iconique de type Macintosh; il offre en outre des possibilités de programmation utilisant le formalisme des acteurs.

Selon les principes d'ergonomie cognitive, la recherche veut faciliter pour l'usager l'apprentissage des systèmes informatiques. En s'inspirant de la méthodologie d'analyse cognitive des processus, elle décrit la situation d'auto-apprentissage du chiffrier électronique, aidé d'une documentation écrite. L'analyse des protocoles verbaux fait ressortir les représentations que se construit le sujet sur la tâche et ses stratégies d'exploration du contenu. Un modèle de la tâche, un modèle de l'usager et un modèle du manuel sont extraits de ces observations; puis ils sont utilisés pour développer un nouveau modèle du tuteur, qui respecte davantage les stratégies et les représentations spontanées de l'usager. Le modèle d'interface proposé s'inspire en outre des recherches sur la compréhension des textes et l'apprentissage des systèmes informatiques. Le système offre des sous-univers d'exercice (training wheel), qui favorisent une appropriation graduelle sans interférences multiples. Le manuel interactif permet ainsi une exploration organisée de définitions, d'exemples et d'exercices et apparaît comme une piste plus intéressante que les manuels ou les systèmes d'aides, pour favoriser l'auto-apprentissage des applications avec interface à manipulation directe.

ENVIRONMENT FOR PROCEDURAL LEARNING ON HYPERCARD

The research is on an intelligent environment for learning procedural literacy of complex computer applications. It suggests a model built with HYPERCARD, for the learning of an electronic spreadsheet. HYPERCARD is used since it allows rapid prototyping of iconic interface, with actor's programming facilities.

In the field of cognitive ergonomy, the research tries to facilitate the learning of computer systems. Using cognitive processes analysis techniques, it describes the "learn alone" situation of a spreadsheet, using a manual. Protocol analysis shows the representations users are developing and strategies they use spontaneously. A model of the task, of the user and of the manual, are extracted; these have served to develop a model of didactic intervention more suited to spontaneous representation and strategies of the user. The model is also drawing from research on text comprehension and research on the learning of computer systems. The systems offer "training wheels" environments (minimizing interferences), that favor gradual appropriation of aspects of the system. This interactive manual allows an organized search for information, offering definitions, examples and exercises and seems a promising alternative to a written manual or to help systems, for the auto learning of complex applications with direct manipulation interfaces.

Environnement pour l'apprentissage procédural sur Hypercard

Aude Dufresne

Département de Communication
Université de Montréal

La recherche s'inscrit dans la problématique de l'enseignement intelligent d'habiletés liées à l'utilisation de l'ordinateur. Dans le domaine de l'apprentissage des ordinateurs, l'amélioration des interfaces permet de plus en plus un auto-apprentissage, assisté de la documentation. Cependant pour ce qui est des applications complexes comme le chiffrier électronique, l'usager y est souvent dérouté et la documentation n'apparaît pas d'une grande utilité. Diverses alternatives sont suggérées pour introduire l'usager au domaine de connaissance et en particulier les "training wheel" (5, 8), qui sont un environnement réduit pour l'exploration des fonctions du système. La recherche montre qu'on peut utiliser l'environnement Hypercard pour développer un système de ce type, qui se situe entre le manuel interactif et l'environnement de simulation. Le système fournit un environnement d'apprentissage plus dynamique que le manuel et plus structuré que le système seul. Il propose une structure pédagogique, qui s'adapte à l'environnement hypertexte et au contexte de l'apprentissage de procédures.

APPRENTISSAGE DES APPLICATIONS COMPLEXES SUR L'ORDINATEUR

Dans le domaine de l'appropriation de l'informatique et plus particulièrement par rapport au phénomène de l'auto-apprentissage sur micro-ordinateur, on constate la difficulté pour les usagers de dépasser un premier niveau d'expertise, afin de développer des applications complexes. L'étude des interactions humain-ordinateur montre que l'apprentissage d'un système informatisé suppose la transcription des intentions de l'utilisateur dans les termes acceptés par le système. Ceci ne constitue pas une tâche triviale, plus les systèmes symboliques sont distincts et complexes, plus difficile est la conception et l'évaluation des opérations. Norman (16) décrit différents niveaux de difficulté: difficulté à identifier les correspondances entre les deux systèmes symboliques; difficulté à contrôler (dextérité, mémorisation); difficulté à évaluer ce qui s'est passé.

Les interfaces à manipulation directe (de type Macintosh) visaient à favoriser l'apprentissage des systèmes informatiques, en rendant plus évidentes physiquement, plus transparentes, plus immédiates, plus réversibles les opérations effectuées par l'usager ou le système (Schneiderman 17). Cependant l'usager est facilement dérouté, lorsque les opérations à effectuer deviennent plus complexes comme dans le cas du chiffrier: les concepts sont plus abstraits (zone de critère); il doit apprendre à planifier, à poser plusieurs actions avant d'avoir un feedback du système; les messages d'erreurs et d'aides qui sont conçus pour morceler les connaissances sont difficiles à comprendre. Parallèlement, la documentation écrite fournie avec le système n'offre pas un support à l'information, qui favorise l'apprentissage.

La pré-expériementation et les recherches (14) ont ainsi fait ressortir les difficultés rencontrées par les usagers dans l'apprentissage des fonctions complexes existant sur le chiffrier électronique. Nous avons utilisé l'analyse des protocoles verbaux (1,3,7,11,15) afin de cerner les diverses phases du processus d'interaction avec le système et la documentation. Ceci a permis de décrire les connaissances sémantiques (13) à acquérir, de même les connaissances explicitées dans le manuel écrit et celles comprises par le sujet. Certains obstacles épistémologiques ont ainsi été mis en relief (manque de feedback, termes ambigues), de même que des stratégies plus efficaces pour l'explication des connaissances au sujet (lexique, ordre générique des définitions, ordre des interventions didactiques, etc.).

RECHERCHES SUR LES ENVIRONNEMENTS D'APPRENTISSAGE DE SYSTÈMES INFORMATIQUES

A)DOCUMENTATION INFORMATIQUE

La littérature sur la compréhension des textes, reliés à l'instruction, suggère diverses mesures, pour faciliter la lecture, la compréhension et la mémorisation des informations: organisation en spira-

le, mise en relief des thèmes (18) favorisant la localisation, limite du nombre d'éléments nouveaux, reformulations, utilisation d'analogies, éléments mnémotechniques, résumés, etc.. D'autres éléments, comme un vocabulaire consistant et un nombre restreint d'éléments d'information, sont également importants (4,9,10, 12).

Dans le domaine de l'apprentissage des logiciels, la documentation écrite a été comparée à une documentation sur écran: la première a l'avantage d'être plus facile à lire, mais n'offre pas les multiples avantages de l'interaction (2):
- figures ayant une composante dynamique
- accès plus flexible aux informations et à la forme didactique
-possibilité de laisser l'usager manipuler ou faire des inférences
- possibilité d'ajuter le niveau de détail
- possibilité d'évaluer l'apprentissage et d'ajouter des fonctions intelligentes, réorientant l'usager
Le modèle d'intervention proposé puise également dans une recherche intéressante de Black et al. (4) sur la documentation des logiciels, qui fait ressortir que l'apprentissage des faits est favorisé par un manuel de forme "squelettique" et encourageant la "répétition"; alors que, pour l'apprentissage des séquences d'actions, un manuel qui amène le sujet à faire des "inférences" favorisera davantage l'apprentissage et la capacité de transfert. Il semble important de développer des prototypes, qui suggèrent des façons de réaliser, d'intégrer ces différents principes en des modèles complets et prescriptifs (9).

B) MODÈLES D'ENSEIGNEMENTS ASSISTE PAR ORDINATEUR

On peut distinguer divers courants dans les recherches sur les systèmes d'enseignements assistés sur ordinateurs; sans les décrire en détail. D'abord les systèmes d'EAO offrent un exposé des connaissances, que l'apprenant traverse; les évaluations sont en général basées sur la mémorisation des concepts vus et comporte peu de dimension active. Ce type de système ne peut être recommandé pour l'apprentissage de procédures, l'étudiant n'étant pas entraîné à faire des inférences et à explorer le système. Les systèmes d'EIAO ajoutent habituellement à cela, une dimension d'interprétation des erreurs et des stratégies réorientant l'usager vers de nouvelles connaissances; celles-ci pouvant être organisées en réseaux (Goldstein, 12), que l'usager explore. Cependant, la dimension de simulation et d'application des connaissances par l'usager y est encore réduite, ce qui limite donc les possibilités d'utilisation pour l'apprentissage des prodécures. D'autres systèmes fournissent un environnement d'exploration à l'usager (Brown et al, 12), un micro monde où l'usager apprend à jouer avec les lois d'un système. La dimension didactique de ce type est en général limitée: support de manipulation directe, gradation des éléments introduits. Ce type de système a été développé pour l'apprentissage de lois physiques ou mathématiques et de façon très limitée pour l'apprentissage de la programmation. Cependant, ces environnements sont limités si les opérations sont imbriquées et complexes et s'il y a plusieurs possibilités de solution.

Les interfaces de planification favorisent l'apprentissage dans les domaines nécessitant des élaborations plus complexes (apprentissage de la programmation, rédaction de textes plus longs). Dans ces cas en effet, les opérations de l'usager supposant elles-même une hiérarchie de buts et de sous-buts, il est apparu important de fournir à celui-ci un outil lui permettant de manipuler d'abord le niveau supérieur des plans, pour ensuite passer à une élaboration plus détaillée (Miller, 18). Selon DiSessa (16), il faut faciliter le processus de programmation en insérant dans cette activité des aspects de manipulation et de hiérarchie. Weiser et Shertz (20) ont montré que les connaissances des experts ne se situent pas au niveau des commandes mais à un niveau plus général des processus; en ce sens, les systèmes hiérarchiques accélèrent certainement l'élaboration d'une expertise chez les apprenants.
Pour l'apprentissage de la programmation ou d'appliquations complexes, le processus de design procède par des approximations et des transformations successives; l'apprentissage ne peut donc se faire de façon trop structurée, mais doit permettre cette recherche graduelle d'une formulation adéquate. Selon Brown (6) il faut chercher à développer des environnements d'apprentissage plus généraux, qui utilisent l'ordinateur pour mémoriser et organiser l'information, et s'en servir pour faciliter la représentation des concepts et leur organisation, pour inciter à l'exploration et pour faciliter la correction des erreurs, en fournissant à l'usager un reflet de ses processus. En ce sens, il est intéressant de voir comment un environnement général comme celui de l'hypertexte peut être utilisé pour faciliter l'apprentissage d'habiletés procédurales.

Ainsi inspiré de ces recherches, le prototype créé devrait offrir les caractéristiques suivantes:

- une dimension hiérarchique favoriant la compréhension du contexte, de même qu'un accès aux détails contrôlé par l'usager
- une dimension dynamique favorisant la compréhension des séquences et des propriétés temporelles des concepts
- une dimension opératoire (devant être insérée dans la hiérarchie), permettant à l'usager d'appliquer les connaissances apprises et de faire des inférences, sur une partie réduite des connaissances (training wheel), pour lesquelles il est alors plus facile d'interpréter les erreurs.

MODÈLE D'INTERFACE SUR HYPERCARD

Le système a été développé sur un sous ensemble d'opérations non-triviales du chiffrier, soit les fonctions de base de données (recherche et extraction d'informations selon un critère).

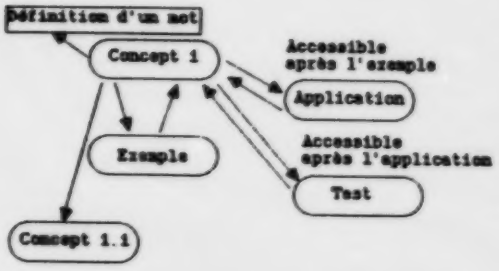


FIGURE 1- STRUCTURE D'INTERACTION

Le système est composé de cinq formats de présentation des connaissances, distinguant, comme le suggèrent Black et al.(4), l'apprentissage des concepts de celui des séquences d'action:
- La page concept, qui donne une définition succinte et donne dans la majorité des cas accès à un exemple figuratif, qui peut-être animé
- La page d'exemple, où le concept est illustré, l'usager restant passif
- La page d'application où l'usager apprend à poser des séquences d'actions, guidé par le système (figure 2)
- La page test de concepts, où l'usager doit répondre à des questions sur ses connaissances
- La page test de séquences, où l'usager doit générer une séquence d'action, pour laquelle il re oit un feedback, qui reproduit et explique les messages d'erreurs du système.

Diverses formes d'accès à l'information sont offertes à l'usager (alphabétique, historique), mais en particulier il favorise l'accès hiérarchique en présentant à l'usager un schéma général des opérations où il peut cliquer sur un concept; il a alors une définition où il est orienté vers une branche d'explications, plus complète, qui peut à son tour conduire à d'autres branches. Les pages concepts forment la structure hiérarchique d'accès, on y offre le choix d'accéder aux exemples, aux pages d'applications ou aux tests.

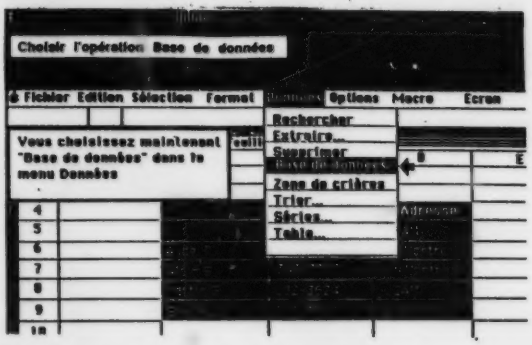


FIGURE 2- EXEMPLE D'APPLICATION

La pré-expérimentation du système a montré qu'il était important de limiter et de gérer l'accès aux connaissances, en limitant les choix possibles et en n'offrant à l'usager que les activités ou les définitions qu'il peut comprendre, par exemple celui-ci n'a pas accès aux tests s'il n'a pas fait les exercices. L'accès hiérarchique est ainsi organisé de façon à favoriser un apprentissage en spirale, les exercices et les figures ne comportent que des éléments expliqués. Pour ce qui est des situations d'inférences, où l'usager doit pratiquer ses connaissances (test des séquences), elles portent sur des sous-ensembles de connaissances; en ce sens, le manuel entre dans la catégorie des "training wheel" (8), qui apparaît de plus en plus dans le domaine des communications humain-ordinateur, comme un moyen efficace d'encourager la manipulation et l'apprentissage (seules les fonctions que l'usager apprend sont simulées, il ne peut donc,par erreur, enclencher d'autres processus).

Le système fournit un outil intéressant pour la recherche, en effet, l'analyse des processus sera utilisée pour évaluer le cheminement de l'usager dans le système organisé des connaissances, de façon à mettre à jour les fonctions les plus utilisées et d'éventuels obstacles à l'apprentissage.

BIBLIOGRAPHIE

1. Aaronson, Amy, Carroll, John M. (1987) Intelligent help in a one-shot dialog: a protocol study. Carroll, John P. et Tanner, Peter P. CHI GI 1987 Conference proceedings. ACM: New York. Pages 183-168.
2. Badre, Albert et Shneiderman, Ben. (1982) Directions in human/computer interaction. Allen, Robert B. Cognitive factors in human interaction with computers. Ablex Publ. Corp.: Norwood, New Jersey.

3. Bailey, Wayne A. et Kay, Edwin J. (1987) Structural Analysis of verbal data. Carroll, John P. et Tanner Peter P. CHI GI 1987 Conference proceedings. ACM: New York. Pages 297-301.
4. Black, John B., Carroll, John M, McGuigan, Stuart M. (1987) What kind of minimal instruction manual is the most effective in CHI GI 1987 Conference Proceedings,. ACM: New York
5. Bonar, J.G. et Cunningham, R. (1988) Intelligent Tutoring with Intermediate Representation. Intelligent Tutorial Systems (ITS-88), ACM, Montréal.
6. Brown, J.S. (1983). Process versus product: a perspective on tools for communal and informal electronic learning. The Educational Broadcasting Corporation., Palo Alto Cal.
7. Card S.K., Moran T.P. et Newell, A. (1983) The psychology of Human-Computer Interaction. Lawrence Eribaum Associates, Hillsdale, N.J..
8. Catrambone, Richard, et Carroll, John M. (1987) Learning a word processing system with training wheels and guided exploration. Carroll, John M. et Tanner, Peter P. Human Factors in computing systems and graphics interface. ACM: New York. Pages 169-174
9. Hartley, James. (1985) Designing instructional text. Kogan Page, Nichols: London, New York.
10. Jonassen, David H. ed. (1982). The Technology of Text: Principles for structuring, designing and displaying text. Educational Technology, Englewood Cliffs, N.J..
11. Kellogg,Wendy et Breen, Timothy J. Evaluating user and system models: applying scaling techniques to problems in human-computer interaction. (1987). Carroll, John M. et Tanner, Peter P. CHI GI 1987 Conference proceedings. ACM: New York Pages 303-308.
12. Kintsch, Walter et Van Dijk, Teun A. (1978) Toward a model of text comprehension and production. Psychological Review, 85,5, p.363-394.
13. Lindsay, P.H. et Norman D.SA. (1980) Traitement Humain de l'information (traduit de l'anglais). Etudes Vivantes, Montréal
14. Marion, G. Vernard, P., Goyer, P., Lemire, L. LeVan, C. Plaisent, M. Sainfort, F. (1985) Observations systématiques de l'apprentissage du chiffrier électronique, rapport présenté à la direction générale des moyens d'enseignement et de la recherche universitaire (DGERU), Ministère de l'éducation du Québec, Québec.
15. Newell, A. et Simon, H.A. (1972). Human Problem Solving. Prentice Hall, Englewood Cliffs, N.J..
16. Norman, D.A. et Draper S.W. (Eds.): User Centered System Design: New Perspectives on Human-Computer Interaction Eribaum, Hillsdale, New-Jersey
17. Schneiderman, B. (1980). Interactive interface issues. Software Psychology: Human Factors in Computer and Information Systems, (p.216-251) Winthrop, Cambridge, Mass.
18. Schwarz, M.K. et Flammer A. (1981) Text Structuring and Title-Effect on comprehension and recall. Journal of Verbal Learning and Behavior, 20, 61-66
19. Sleeman, D and J.S. Brown (Eds): Intelligent Tutoring Systems, (pp. 227-282). Academic Press, New-York.
20. Weiser, M., Shertz, J. (1983). Programming problem representation in novice and expert programmers. International Journal of Man-Machine Studies, 19, 391-398.

HYPERTEXT AS A MEDIUM FOR INTERACTIVE LEARNING

M.R. Kibby, J.T. Mayes, C. Knussen
Scottish HCI Centre
University of Strathclyde
Glasgow, UK

(Abstract Not Available at Time of Printing)

HYPERTEXT EN TANT QU'OUTIL D'APPRENTISSAGE INTERACTIF

(Résumé non disponible)

HYPERTEXT AS A MEDIUM FOR INTERACTIVE LEARNING

M.R. Kibby, J.T. Mayes and C. Knussen

Scottish HCI Centre, University of Strathclyde, 36 North
Hanover Street, Glasgow G1 2AD, Scotland, UK

Our involvement with hypertext came in 1984. We had looked at our first Macintosh computer and had immediately recognised that it represented an important step forward in human-computer interaction and computer usability. We went on to examine how learners could and should learn from such interactive systems and identified the main factors impeding the application of computer based learning on a wide scale: the stultifying constraints of existing computer systems and the high cost of developing quality learning materials. Our dissatisfaction with the existing technology led us to take a number of initiatives as our ideas evolved and we came to realise we were, and are, in the midst of a world-wide surge of interest in hypertext.

The three projects described here cover different aspects of hypertext systems but point in the same direction. We built a hypertext-like system for tutoring on the Macintosh well before the advent of *HyperCard* [1]; we are training teachers to use*HyperCard* , and we are partners in a new European Community project devising a hypertext learning system for the 1990s. Our preliminary conclusion is that hypertext is powerful and has wide potential application in various learning environments, but we are not yet sure that organisations delivering learning and training are able to accommodate a switch on a large scale to the interactive computer based methods exemplified by hypertext.

The StrathTutor Project

Our first project was funded by the U.K. Computers in Teaching Initiative of the University Grants Committee. Starting in late 1985 we built and tested an interactive tutorial system shell called *StrathTutor* based on the discovery learning principle. No programming was required, and therefore there was no new programming language to learn. Since *StrathTutor* has a number of other unique features for a hypertext system, it will be described in some detail.

StrathTutor codes knowlege by requiring the author to index each frame, like earlier attempts such as SMITH [2]. However, it departs radically from previous systems by requiring no coding at all at the level of the organisation of the knowledge. It codes no links between frames, either directly or in the form of sequencing rules. Neither is any modelling of the student's developing understanding attempted and in that sense *StrathTutor* is not a *tutoring* system at all. Instead, *StrathTutor* attempts to provide a maximally supportive environment for interactive learning. That is not to say that it lacks a pedagogic stategy, however. On the contrary, we regard it as an example of a reactive learning environment [3]. Unlike a conventional hypertext system, for example *Hitch-Hikers Guide* [4], where there are explicit links between frames of information, *StrathTutor* supplies the links by computing them from the associated knowledge base while the program is running. It remains true hypertext because the material appears linked to the user.

Important features of *StrathTutor* are:

• Knowledge is separated from the contents of individual frames and represented as attributes; associations between related frames are generated by the use of pattern matching heuristics. Areas of interest within a frame are designated as transparent buttons by the author and allocated attributes. Links are computed at run time by pattern matching heuristics, according to the type of interaction which the learner initiates. Some of the heuristics are derived from simulation models of human memory [5].

• The learner can explore the knowledge base in a number of different ways.
(1) From a 'Progress' menu . As each frame is displayed an algorithm determines and displays the titles of the three most related frames from those still unseen.
(2) By frame title.
(3) By following a guided tour which will allow the learner to see all frames that are coded with the combination of attributes that the learner has selected.
(4) By selecting a button as requiring further investigation.

• Complete learner control. Progress through the material is entirely a matter of choice by the learner. Browsing in *StrathTutor* is largely conceptual; the learner is required continually to search for links between frames at the level of the underlying concepts (attributes).

• Computer-generated self-tests. The 'Quiz' menu allows the learner to attempt to identify the buttons on two frames that have greatest similarity in terms of attributes. Feedback is given in the form of a game score and, if requested, a 'best answer'.

A fuller account of StrathTutor and its initial evaluation is given in reference 1. Attempts to evaluate our approach to the construction and presentation of learning material have centred on two methods.

The first approach is the analysis of tracefiles generated automatically each time the learning system is used. Such tracefiles analyse the learner's choice of progress from one frame to the next, the time spent on each frame, the setting of a number of user-controllable characteristics, the use and success the learner has with the quiz options. Pre- and post-tests were administered to 17 subjects who were undergraduate and postgraduate students with little knowledge of the test domain, glaciation. There was a significant difference between subjects' pre- and post-test scores, and a number of interactions between variables, greater than expected by chance, were detected. However, these may not be causal relationships and it was difficult to draw further conclusions.

The second approach to evaluation has been some preliminary trials with the technique of constructive interaction [6] in which pairs of subjects are videotaped while using the learning system. This has proved to be particularly effective because one subject is required to justify further progress through the material by convincing the other of the appropriate mouse-driven action to take, thereby exteriorising at least some part of the learning processes. Since almost all subjects in the experiments described here were unfamiliar with the Macintosh computer, learning included how to use the Macintosh and *StrathTutor* as well as learning about the subject being studied.

One conclusion from the evaluations is that subjects become so engrossed in the task of learning that they appear to forget to use the full range of navigation tools available to them. This can lead to concentration on only a few modes of use. Our evaluations indicate that subjects certainly learn from the system, and that it provokes much learning-centred discussion when used by pairs of subjects. This confirms subjective observations made during class use.

Authoring

We expected at the time that most tutorials for *StrathTutor* would be written by colleagues in the Department of Biosciences who we imagined would see the long-term advantages, for themselves as well as for the students and the Department, of establishing a substantial body of computer based material for undergraduate teaching. We did realise, however, that this was unlikely to happen without a considerable effort on our part to spell out these advantages and to make the other staff feel that this was not just a personal research grant but was a Department-wide project in which they each had a role to play. To this end we held an afternoon seminar to which all staff were invited, we issued a number of news-letters explaining progress on the project, we held a series of discussions with many individual members of staff, and we held an open day in the Department at which all software available with potential use in teaching was demonstrated.

The response was patchy. A number of promises were made and in one or two cases some real enthusiasm was expressed but from the majority of staff there was no response at all. Nevertheless, we still had an important card to play. We offered staff who would undertake to write a tutorial based on their own teaching material a Macintosh and software on their desk for unrestricted use for as long as we could allow, given continuing commitments to teaching. This offer was taken up by eight staff. While a few tutorials have now been written by departmental staff other than those deeply involved with the project, a number of those who accepted our offer still failed to keep their side of the bargain.

We asked whether there are factors specific to this project that are responsible for the disappointing response from other members of staff. Accordingly, one of us, not a member of the Bioscience Dept. (J.T.M.) conducted interviews with members of the Department specifically to elicit views about the project and about personal involvement. It was apparent that knowledge about the aims and objectives of the project varied considerably, as did views about the potential benefits of developing educational computing within the Department. On one point all were agreed. The general climate within U.K. Universities did not at present reward efforts towards innovation in teaching. One comment from a staff member who had contributed well above average to the project (a senior member of the Department, currently nearing retirement) illustrates this point. He remarked that he only felt able to contribute time and interest because he "no longer had anything to lose".

The evaluation of HyperCard in the hands of teachers

The next project evolved out of *StrathTutor* when *HyperCard* was introduced for the Macintosh computer, for there are similarities in approach between the two systems. Indeed, *StrathTutor* has been incorporated into *HyperCard* and delivered to the project, which commenced in June 1988, and which is funded by the Scottish Council for Educational Technology. The principal aim of the project is to evaluate both forms of hypertext as computer based learning systems for the school environment. This evaluation includes both authoring and learning modes, using both teachers and children. The project is approaching the end of its first year and has addressed the question of training teachers in *HyperCard* and the production of high quality teaching materials in *HyperCard* for use in schools. This paper records the experiences and decisions taken during the design of training materials for the evaluation proper later in 1989.

Preliminary data of schools and teachers participating in the project were obtained from short interviews with head teachers and individual teachers. Substantial variation were revealed in such factors as availability of hardware and software, school and departmental policy regarding the use of computers in teaching, personal experience and confidence of using computers in teaching, attitudes towards computers, knowledge and skill relating to the use of computers, and experience of using the Macintosh and *HyperCard*. The teachers were not a random sample, but were suggested by various computer advisers as being interested, but not necessarily experienced, in using computers in teaching.

Given the variation in computing skills and experience found in individuals, it was necessary to devise a method of training in *HyperCard* which could accommodate individuals of different background and take into account the fact that training must take place in the individual's own time. *HyperCard* is accompanied by an on-line Help stack and an instruction manual, descriptive material is circulated by Apple to advice centres for schools, and there is a good reference book

available [7]. However, these did not appear specific enough for the aims of the project. Preliminary discussions with teachers and computer advisers pointed to a general preference for training materials grounded in clear instruction rather than exploration. As one participant put it, "Unguided discovery is a luxury teachers can't afford". (This phenomenon may yet be the nemesis for discovery learning systems in formal teaching environments.)

Carroll *et al.* [8] noted that self-instruction manuals for novices in computer applications often conflict with common learning styles. They recommended that training materials should focus on early achievement of practical skills, they should provide adequate information to promote problem-solving, and documentation should be kept to a minimum. The development of training materials for this project was based on these principles, using *HyperCard* itself as the medium for training.

The initial plan had been to produce five units based on the Macintosh and the four*HyperCard* user levels, apart from scripting, but two units had to be allowed to cover the production of graphics. The use of user levels ensured that skills and concepts were introduced when it was possible to practice with them. Development of training materials went through several cycles of testing in the hands of teachers followed by revision during the pilot study.

During the first cycle, questionnaires and interview schedules were developed to access the themes investigated by Chomienne [9] relating to individual teachers: computing experience, teaching experience, use of computers in the classroom, and attitudes towards computers and teaching. It emerged that the interviews elicited more information than the questionnaires, and that the information was more detailed and of better quality. In contrast, many items in the questionnaires were omitted by participants or only partly completed and there were complaints about paperwork. As a result, the interview schedule was adapted to cover all questionnaire items and used alone in the second stage of the pilot study, and more guidance in using a revised log book was provided.

The most important finding of the second cycle concerned the training materials themselves. Although revisions had improved them, all participants found the style of presentation was still ambiguous and not sufficiently didactic and the *HyperCard* examples were judged to be too heavily dependent on the documentation, which had to be read to understand screen events. Further revision was carried out before the materials were prepared for the evaluation proper (to be performed from March to May 1989).

On being asked if they would be prepared to spend time authoring their own stacks for use in the classroom, a few teachers out of 30 in the study said that they might. Most said that they would, depending on the time it was going to take, and how easy it was going to be. No one said yes without qualification.

More information was gathered from the eight teachers interviewed as part of the pilot study. Reactions to *HyperCard* appeared to be influenced by the following factors, which tend to confounded with attitudes to the introduction of computers *per se:*

- The hardware situation in the school and in the local administrative area, both at present and in the future.
- The location of Macintoshes in the school and the arrangements for getting access to the hardware.
- Prior experience of using computers in the classroom. This includes experience of software, and is partly dependent on the subject matter taught by the teacher. This reflects the general feeling among those with no experience of *HyperCard* that software is developed by other people rather than being under one's own control.
- The teacher's own views on the role of computers in schools. The minority view that computers should really be kept for computing classes may be under- represented in our sample because the participants were drawn from those showing special interest in computers.
- The goals of the teacher's own teaching strategies. The teachers most open-minded in their reaction to *HyperCard* were those who believed that learning should be an active, not a passive, activity, and those trying to encourage pupils to take responsibility for their own work.
- The teacher's career goals. The introduction of computers into schools in the region has produced many opportunities for teachers willing to put effort into learning to use computers. Such teachers were more likely to be willing to learn to use *HyperCard.*

Some of the attitudes expressed here overlap with the management and training issues identified by Chandra *et al.* [10] and Chomienne [9]. However a greater incidence of personal factors was reported in this study, possibly because teachers were using computers outside the school organisation.

A few teachers had ideas for *HyperCard* stacks. Some teachers became more enthusiastic about *HyperCard* when it became clear that images and sound could be incorporated easily into stacks. However, of the minority of those having seen a *HyperCard* stack, very few had any insight into the methods of constructing a stack. In general, it seemed that teachers did not see themselves as authors of stacks. Moreover, they did not seem to relate to the idea that in *HyperCard* it is possible to adapt other software for one's own use, even though most had complained that commercial software was not usually specific enough to be useful. One problem is clearly the current presentation of *HyperCard* as a personal organiser - which does not interest teachers.

Few participants were willing to commit themselves to the production of teaching materials for use in the second year of the project, understandable given their general unfamiliarity with *HyperCard* and the perceived need for computer graphics skills to design and execute aesthetically pleasing screens (which most teachers thought they lacked). However, this attitude does call into question the degree of interactivity with teaching materials which teachers at this time envisage for themselves.

The DELTA programme

DELTA is a European Community programme aimed at Developing European Learning through Technology Advance. The current phase of the programme, commencing in 1989, is exploratory. DELTA has several objectives:

- encouraging the emergence of a cohesive industry in advanced educational technology within the Community;
- common learning support environments and advanced telecommunications;
- concerted effort by administrations, educators, software and hardware industry, advanced publishers;
- changing the way the main actors think about learning [11].

The project in which we are partners includes Philips, the electronics group, Argus, a subsidiary of the Elsevier publishing house, and various other European companies and universities. We are particularly involved in aspects of future hypermedia systems and the demands an interactive, exploratory learning environment will make on the hardware and software. We are not constrained to consider formal learning environments as they exist in our educational system or in company-based training establishments — open and distance learning are regarded as just as important in the DELTA programme. Our approach at this stage, is not to attempt to develop a fully integrated system using existing technologies, but to isolate those aspects which a future development programme will require to address to be able to advance significantly on present implementations of hypermedia. The research will focus on 'islands' where our knowledge is lacking such as hypermedia databases, knowledge representation, the usability of hypermedia systems, and navigation problems which various workers have observed and reported [12].

At this stage it would be premature to speculate on the outcome of this project. What the presence of it and other hypermedia oriented projects in DELTA indicate is that hypertext and hypermedia are being taken very seriously as future modalities of learning.

Conclusion

Our experience with *StrathTutor* and *HyperCard*, from designing software and courseware to the training and management of users is that our educational institutions are not able to utilise new technology effectively. Either hypertext and hypermedia based interactive learning systems will pass them by or they will be driven by its potential to deliver more efficient and more effective learning to adapt to its requirements. In the latter case it will demand of all actors in the world of learning must that they be ready to re-appraise their role in the delivery of education and training.

Electronics companies, producers of audio CD, will discover a potentially huge education market for inexpensive CD-ROMs; authors will have to think about hypertext versions of their books; creators of courseware, graphic designers and typographers will have to set themselves different and even higher standards to suit the new form; the publisher will have to be able to support various presentations of a basic text to serve various types of user; schools and colleges will have to be prepared to deliver learning in various forms to suit the requirements of learners more than to serve administrative convenience; teachers will have to be able and enabled to promote learning in ways which hitherto impossible to deliver; learners must learn to take far more responsibility for their learning.

The education community is not yet alert to the nature of interactive learning with hypertext and the consequences which will ensue from its large scale adoption.

References

1 Mayes J T, Kibby M R and Watson H, *StrathTutor*: the development and evaluation of a learning-by-browsing system on the Macintosh. *Computers and Education* 12, 221-229 (1988).

2 Osin L, SMITH: how to produce CAI courses without programming. *Int. J. Man-Mach. Stud.* 8, 207-241 (1976).

3 Brown J S, Learning by doing revisited for electronic learning environments. In *The Future of Electronic Learning* (White M A ed.). Lawrence Erlbaum, N.J. (1983).

4 Hammond N and Allinson L, Development and evaluation of a CAL system for non-formal domains: the Hitch-Hikers Guide to cognition. *Computers and Education* 12, 215-220 (1988)

5 Hintzman D L, MINERVA 2: a simulation model of human memory. *Behaviour Res. Meth. Instrum. Comput.* 16, 96-101 (1984).

6 O'Malley C E, Draper, S W and Riley M S, Constructive interaction: a method for studying human-computer interaction. In *Human-computer interaction* (Edited by Shackel B) North-Holland, Amsterdam (1985).

7 Goodman D, *The complete HyperCard handbook*. Bantam, Toronto (1987).

8 Carroll J M, Smith-Kerker S L, Ford J R, and Mazur-Rimetz S A, The Minimal Manual. *Human-Computer Interaction*, 3, 123-153 (1988).

9 Chomienne M, Educational computing implementation: the case of Quebec. *Computers and Education*, 12, 57-61 (1988).

10 Chandra P, Bliss J, and Cox M, Introducing computers into a school — management issues. *Computers and Education*, 12, (1988)

11 *Information package for the DELTA call for proposals, 1988*. Commission of the European Communities, Brussels.

12 Hardman L, The human hypertext interface. In R McAleese (ed.), *Hypertext 1*, Ablex, Norwood, New Jersey. In the press.

IMPLEMENTATION ISSUES/

QUESTIONS DE MISE EN OEUVRE

REVIEW OF THE USE OF INFORMATION TECHNOLOGIES IN ONTARIO SCHOOLS

L.R. Smith
Ministry of Education
Toronto, ON

Since April 1988 the author has conducted an Information Technologies Review for the Ontario Ministry of Education. During this review he visited many Ontario schools and classrooms to observe the use of a wide variety of information technologies, including the textbook, video, television, film, computer, electronic communication, CD-ROM, and videodisc. A summary of the observations, literature search, levels of use and problems will be reported. As well, a Provincial policy framework for the support of integrated use of these technologies to enhance the learning of students in elementary and secondary schools will be outlined. The author will link this framework to the existing policies of the Ministry of Education and school boards. The session will include many anecdotes of teacher and student use of the technologies both effectively and ineffectively.

EXAMEN DE L'UTILISATION DES TECHNOLOGIES D'INFORMATION DANS LES ÉCOLES EN ONTARIO

En avril 1988, l'auteur a commencé une étude sur les technologies d'information pour le Ministère de l'Éducation de l'Ontario. Au cours de cette étude, il a visité plusieurs écoles en Ontario pour y observer l'utilisation d'un grand nombre de moyens d'information : livre, bande magnétique, télévision, film, ordinateur, communications électroniques, disques compact et vidéo. Un sommaire des observations, de l'exploration de la littérature, de l'utilisation et des problèmes sera présenté. Les grandes lignes de la politique du gouvernement provincial pour venir en aide aux utilisateurs de ces technologies pour améliorer l'enseignement dans les écoles élémentaires et secondaires seront également présentées. L'auteur établira le lien entre cette politique et les politiques actuelles du Ministère de l'Éducation et des conseils scolaires. La présentation comportera de nombreuses anecdotes mettant en cause des enseignants et des élèves et illustrant les façons efficaces et non efficace d'utiliser ces technologies.

REVIEW OF THE USE OF INFORMATION TECHNOLOGIES IN ONTARIO SCHOOLS

LORNE R. SMITH

Ontario Ministry of Education

INTRODUCTION

Since April, 1988 the author has been reviewing the use of information related technologies including television, video(VCR), microcomputers, electronic communications and optical storage devices and their associated French and English learning materials in Ontario elementary and secondary education.

METHODOLOGY OF THE REVIEW

The Review has systematically collected information from school boards, schools, Faculties of Education and leading technical and educational experts through interviews, observations and questionnaires. The relevant literature and Ministry Policy documents have also been reviewed.

CURRENT CURRICULUM POLICY

Ministry curriculum policy identifies the need for creating a teaching/learning environment where information, problem solving, communication, and cooperative learning are integrated to help students develop the skills they will need to be successful and productive citizens.

The Geography Curriculum Guideline, Intermediate and Senior Division, 1988, demonstrates the need for up-to-date information from many sources:

> "Traditional textbooks cannot provide the current information that geographic studies demand. In partnership with the teacher-librarian, geography teachers should ensure that students become increasingly capable of locating and using up-to-date information in a wide variety of forms including audio-visual material."

An active learning environment that requires the use of information in an integrated manner and helps students maintain their curiosity is an essential component of the Junior Division program. Ages 9 Through 12, A support Document to The Formative Years, 1986 indicates:

> "The intent of this document is to provide strategies across curriculum areas to assist teachers in designing an active learning environment for whole-class involvement, small-group activities, and individual pursuits. In this environment children are encouraged to investigate, think, judge, appreciate, imagine, feel, enjoy and create".
>
> "Learning is something children do, not something that is done to them. It is a multi-sensory process, which requires children's active participation in seeking and using information in an integrated manner in order to arrive at meaning."

INFORMATION TECHNOLOGIES FRAMEWORK

Figure one is a Framework to assist in organizing thinking about information/resource based education and the role the computer and other technologies can have in supporting it.

A major activity of schools is the Production or Publishing of materials. Students are given an assignment to produce a paper, write a test, produce a video, give a speech or create some type of visual. At the present time many of these end products never go beyond the student and teacher. However many electronic tools are now becoming available to assist students to improve the quality of their final products. These include word processors, video editors, and electronic music systems. A quote from a student produced newsletter in Northern Ontario exemplifies the value of the wider circulation of these finished products: "Finally our paragraphs are going to be read by real people, not attacked by a teacher's red pen". The right hand column of Figure one summarizes these activities.

To produce such products, students must have access to a wide range of information. These may be primary sources such as personally collected information from interviews or observations, dictionaries, periodicals, data bases, speeches, videos or from artifact tools developed by other people. These include textbooks, documentaries, computer software, kits, etc. Most of the information used in schools is print based but the importance of

Figure one

information in audio and image form is growing as the technologies to capture, store and replay these forms improve. The two left columns on Figure one summarize these sources.

Materials are of little value without the intellectual activity of analyzing and synthesizing the information into a meaningful product. Collaborative talk is an important part of this process. The THINKING/PROCESSING of the INFORMATION before PRODUCING/PUBLISHING the final product represents the LEARNING of the student. These higher order thinking skills should take precedence over the more technical skills of accessing the information or producing the final products. These activities are summarized in the THINKING/PROCESSING column of Figure one.

A fundamental premise of the model represented by Figure one is that there is as much intellectual activity(LEARNING) in the production of a quality video, piece of music or a speech as there is in the writing of a story or a term paper.

Demand for quality work by teachers will ensure that the THINKING/PROCESSING SKILLS continue to be sharpened and enhanced learning takes place.

FINDINGS FROM VISITS

The following findings have been synthesized from information collected between May, 1988 and January, 1989.

-Schools are using a wide array of information technologies in a wide variety of applications.
-At this time in the evolution of the use of computers, the only application which those interviewed agree has an impact is word processing. Students write more, revise their work and the materials produced look better.
-There is agreement that computers are intrinsically motivating to students, generate more cooperative learning, and improve student self-image.
-There is no agreement as to whether computers should be in labs or in classrooms. Many teachers would like to see both in a school.
-There is no agreement as to the future of Logo.
-Many secondary students are leaving school without any experience on computers.
-There is a very uneven acquisition of software to maximize the benefits of the computers currently in schools. Classroom visits indicate that students using licensed software generally have access to a wider range of software on a daily basis than those in schools where software must be purchased at commercial rates.
-The use of video in classrooms is increasing

-There has been a sharp decline in the purchase of 16mm film.
-Students make excellent use of CD-Rom when it is available. Virtually all materials observed in use are of U. S. origin.
-Copyright is a significant issue. Most schools rely on TVOntario to ensure copyright is enforced on video.
-There is generally a shortage of quality French language materials in comparison to those available in the English language.
-Generally, special education programs have not exploited the use of the technologies to assist learning.
-There is a willingness to make greater use of the technologies in schools but teachers indicate they need more time and help to review and learn to use the technologies and the related materials. Materials must also be readily available and easy to use before general adoption takes place.
-Many schools and teachers wish to explore the power of electronic communication to bring new interest and remote expertise to the classroom.

COMPUTERS IN ONTARIO SCHOOLS

	SEPT/88+
First Generation(8-bit)	56,112
Second Generation(G.E.M.S.)*	34,689
Second Generation(Non-Gems)	10,344
Total	101,145
Av No. Computers per school	22.5
Elementary	14.7
Secondary	61
Student/Computer Ratio	17.6/1

+Preliminary Data
*Grant Eligible Microcomputer Systems meeting Provincial Functional Requirements.

POLICY ISSUES

There are many issues which have been identified by teachers, boards and the Ministry to ensure the effective use an ever increasing variety of information technologies. The following are a few of these issues:

1.ACCESS TO INFORMATION AND MATERIALS
It is evident that to implement the Curriculum Policy of Resource Based Learning it is essential that all students have access to a broad range of information and materials. The majority of new information is being stored electronically(CD-ROM, Data Bases, Videodisc, Video) and students need opportunities to use information in this form. Funding levels are affecting the ability of schools to provide these materials.

2. QUALITY OF MATERIALS
New policies and strategies at the provincial, board and school levels are necessary to ensure materials used in schools are of high quality and are consistent with provincial and board curriculum policies.

3.EVALUATION OF CURRICULUM AND PRACTICE
The computer and the electronic storage of information is creating an entirely new way of thinking about THINKING AND LEARNING. The need to evaluate both the curriculum and teaching practice in classrooms is becoming evident.

4. ASSESSMENT OF PERFORMANCE
The assessment of pupils involved in group activities, resource based project learning, electronic communication and video projects requires a new set of criteria and strategies for measuring successful performance.

5. ORGANIZATION CHANGES
Policies and organizational structures need to be explored to ensure the integration of the various forms of information and their related technologies.

6. HIGHER ORDER THINKING SKILLS
Research, at this time, is inconclusive on the impact of the technologies on student achievement in curriculum areas and problem solving. They can, however, enhance student self concept, and motivation. Teachers need models of curricular based applications of technology that have been shown to enhance higher order thinking skills. New curriculum materials that demonstrate the integration of software, video, print and other technologies are also needed.

7.FUNDING LEVELS
What level of spending is appropriate at the Provincial, Board and school level to ensure students have access to the necessary information and technologies to develop, over the course of their school careers, the skills required to be successful in the information society?

8. COPYRIGHT
There remains considerable confusion in schools about the new copyright legislation. Policies and funding levels must clearly recognize the needs of teachers to have access to a wide range of materials for use in the classroom.

CONCLUSION

Ontario has made major strides in the introduction and use of these new information technologies. There is an overriding sense of optimism in boards and schools about the power of the technology to be a catalyst to improve what goes is classrooms. Students are stimulated because they are finding a world with which they are familiar and one which they can control. The video camera, the music synthesizer, the computer and laser technology are providing exciting new opportunities for children to learn and to be creative. However, much work is still required to ensure the applications are fully integrated into the classroom curriculum.

REFERENCES

1. Owston, R. D. Summary of First Year Observations of Computer Use at Sam Sherratt School, Centre for the Study of Computers in Education, York University, June, 1988.

2. Geography Curriculum Guideline, Intermediate and Senior Division, Part A, Policy and Program Expectations, Ontario Ministry of Education, 1988

3. Ages 9 Through 12, A Resource Book for Teachers, Ontario Ministry of Education, 1986

4. Collis, B. Research Windows, The Computing Teacher, August/September 1988, Volume 16 Number 1, 6-7.

5. Fullan, M.G., Miles, M.B., Anderson, S. E., Strategies for Implementing Microcomputers in Schools: The Ontario Case, Ontario Ministry of Education, 1988

THE "POLITICS" OF INTRODUCING CAL INTO LAW COURSES AND USER PERSPECTIVE ON CAL

M. Young
Dept. of Legal Studies
Coventry Polytechnic
Coventry, UK

There are two main "political" problems to be dealt with when trying to introduce CAL into a course which has been running in a department for several years. In my case I was trying to introduce CAL into the first year of our LL.B degree.

The first problem was gaining the approval of the Head of the Department. The second problem was gaining the approval of the other members of staff within the Department. Having gained the approval of the Head of the Department and the other members of staff one must be prepared to deal with an "adverse" user (student) perspective to CAL. Immediate student reaction to CAL can be most misleading but unfortunately this "adverse" student reaction can easily manifest itself in the form of the Head of the Department having a sudden about turn on the idea of CAL. However, once a teacher has decided to design, write and implement their own CAL tutorials they should not be put off from seeing their project through to the end. They need a fall-back position in case the unexpected happens but should not be pressured into abandoning the project.

LA "POLITIQUE" VISANT À INTRODUIRE L'EAO DANS LES COURS DE DROIT ET POINT DE VUE DE L'UTILISATEUR SUR L'EAO

Il y a deux problèmes "politiques" majeurs que l'on doit essayer de résoudre lorsqu'on veut introduire l'EAO dans un cours qui est donné dans une faculté depuis plusieurs années. En ce qui me concerne, j'ai essayé d'introduire l'EAO dans la première année du cours menant à la licence.

Le premier problème consistait à obtenir l'approbation du doyen de la faculté. Le deuxième problème consistait à obtenir l'approbation des autres membres du personnel dans la faculté. Après avoir obtenu l'approbation du doyen de la faculté et des autres membres du personnel, il restait à aborder le point de vue de l'utilisateur (étudiant) "hostile" à l'EAO. La réaction immédiate de l'étudiant à l'EAO peut être très trompeuse, mais malheureusement cette réaction "hostile" peut facilement entraîner une volte-face soudaine du chef de la faculté relativement à l'EAO. Toutefois, lorsque des professeurs ont décidé d'élaborer, de rédiger et de mettre en oeuvre leur enseignement dirigé (EAO), ils ne devraient pas être empêchés de mener leur projet à terme. Ils ont besoin d'une position de retrait en cas d'imprévu, mais ils ne devraient pas être contraints d'abandonner leur projet.

THE "POLITICS" OF INTRODUCING CAL INTO LAW COURSES AND USER PERSPECTIVE ON CAL

MAX YOUNG

Department of Legal Studies
Coventry Polytechnic
COVENTRY U.K.

INTRODUCTION

At first sight it might appear that this paper deals with two separate things but this is not the case; favourable user perspective of CAL makes its introduction into a course much easier and without favourable user perspective, CAL, however good, will not remain long in a course.

THE CAMPAIGN

To develop the political metaphor no political victory is acheived without careful planning (and a little luck). I see the campaign of introducing CAL into a course in the following steps: 1. the idea; 2. pilot development; 3. student interest; 4. the suggestion; 5. tutorial development; 6. implementation; 7. student feedback and Head's reactions; and 8. Monitoring.

1. The Idea

The most important step in the introduction of CAL into any course is the reason for the incorporation of CAL into that course. To state that CAL, per se, is a good thing, is to loose sight of any educational advantage or merit that really should be the reason for the new teaching medium of CAL. It should always be remembered that when any new teaching method is being considered it is the potential educational gain, (either it is better in some way for students or for staff), that should be the reason for the new teaching method, rather than the novelty or trendiness of the new teaching method or medium.

If, having considered the educational aims and objectives of why CAL tutorials are to be developed you still consider that CAL is the right medium with which to tackle those aims and objectives, you will have a sound foundation for the incorporation of CAL into your course.

2. The Pilot Development

The pilot development stage of any CAL is a most important and valuable stage in the introducation of CAL into a course because not only does it deal with such matters as:

(i) the need to explain to the students the limitation of a computer program which tries to deal with completely free-text responses;

(ii) finding out how much time is involved in preparing CAL tutorials;

(iii) 'help' facilities;

(iv) the need for the programming language to be able to deal with free-text responses;

(v) appropriate screen design and layout; and

(vi) a suitable CAL programming language.

Also if students are actively involved at this stage student interest (see 3. below) can be aroused and gauged.

3. Student Interest

In the early days of the pilot development of my CAL tutorials, before they were formally introduced into the course as part of normal tuition, groups of students ranging in number from 2 to 8 would come along to my room and ask to use the CAL tutorials that they had heard about. This was most encouraging and helped me enormously in the development of the CAL tutorials. I would remain in the room and get on with my work whilst the students would elect one of their number to 'operate' the computer. The students would then discuss in a very full and thorough way each point of the tutorial problem with everyone contributing to the discussion. It was in many ways what an ideal tutorial should consist of with the students leading and exploring the tutorial without any intervention on the part of a lecturer. I pointed out to them the fullness of their discussion even though I was present and their explanation was that they felt free to discuss the problem and then see what the computer came up with as the correct answer. If they disagreed with the computer's answer they felt free to disagree with the computer. This early observational experience led to the speculation that the CAL that I was developing was useful in that, inter alia, it could promote discussion better than NONCAL tutorials. Further, I could see that there was a student demand for the CAL tutorials and in fact students asked if there were more tutorials they could do. This student interest and demand proved most useful when stage 4 - The Suggestion - was undertaken.

4. The Suggestion

Following the stages of conceiving the idea, doing the pilot development, and satisfying myself of student demand for CAL tutorials, the next step was to consider how to go about the incorporation of the CAL tutorials into the course. It might well be that a proposal to 'instantly' introduce CAL into a course would have met much opposition from various quarters. This problem was overcome in three ways. Firstly, introducing change needs support from the management and one important strategy for the successful introduction of change dicates that the Head of Department should be made aware, right from the inception of a CAL idea, of what is intended and, perhaps more importantly, why it has been planned in that way. It also helps at this stage if not only a potential educational gain can be demonstrated but also, especially in these days of scarcer resources, if it can be shown that there is potentially a resource gain to be had by introducting CAL tuition. Despite making the Head of Department fully aware of the various matters perhaps the best that can be hoped for at this stage from most Head's of Departments is tacit approval.

Secondly, another helpful strategy for change can be to undertake the preliminary 'proving trials' of the new material oneself, before its general introduction. In this way there should be no threat to other members of staff in the department.

A third strategy I adopted was, at the same time, to make my colleagues aware of my plans. As long as they did not have to be involved in any of it they seemed fairly happy to let me get on with it. In my case this was especially important since I shared the Contract Law course with a colleague who was not (and still is not) very keen on IT. After some little discussion with my Head of Department and my colleague, and some reassurance from me as to what would happen if things went 'wrong' (this seemed always to be the main concern of my Head of Department) it was agreed that I could proceed with my plan.

5. Tutorial Development

Having had the approval to proceed with the incorporation of CAL into the course, the next and most time-consuming stage was undertaken, viz the development of enough CAL material to make it a significant part of the course. The significance of this stage cannot be understated. A great deal of time is involved in the preparation of CAL tutorials: in my case this was in my own 'spare time' with no timetable allowance given. The alternative of developing only a small amount of CAL material makes it difficult to evaluate because using only a small amount of CAL tends to make it novel which can lead to over generous results in favour of CAL. It is at this point that many people with good ideas flounder because of time restraints. Let me add that if you have a Learning Services Unit at your college their help (moral support) can be invaluable - especially in those black days of gloom when you are ready to give it all up for one reason or another.

Also important at this stage, especially in these days of tight resources, is to negotiate with your Head of Department, before implementing the CAL tutorials, as to what will happen to your 'free time' that is generated when the students are using the CAL tutorials and you are not present; do you end up with an increased teaching load as a consequence of the introduction of the CAL tutorials? This really is a great fear of many people and puts many people off developing CAL material. In my case I negotiated that the 'free time' (i) would be used to monitor the students' progress (see 8. below), (ii) would be some little compensation for the time spent in developing the CAL tutorials and (iii) that in the future the 'free time' might be used to provide remedial tuition for those students who needed it. I would add that there is now a developing tendency to move away from traditional lectures and tutorials to a more flexible approach to teaching/learning whereby each lecturer will be provided with a certain amount of time to 'teach' the course but it is open to the lecturer to run the course in any way he thinks fit. In these circumstances it may be that CAL is one of the most appropriate tools.

6. Implementation

(i) Introduction of students to CAL

Each year in the first week I explained to each group of students, during their normal tutorial time, what a computer was and how it worked. I told them that any problems they might encounter by way of a technical problem were my fault and not the fault of the computer. This was to put them at ease with the computer and to enable them rightly to blame me instead of the computer, if, for example, the computer rejected a correct answer. The explanation of the computer and its working was very simple and totally non-technical. This helped diminish the mystique of the computer and helped put the students at ease. I found from the questions that the students asked about the computer showed that they were somewhat apprehensive. It is still true in the case of our course that the vast majority of students have not used computers before they start on our first year. Also, some students were worried that they would have to program the computer or 'talk' to it in some technical manner.

After I had given my introductory talk to the students each pair of students started up their computer. I then explained to the students the purpose and use of the CAL tutorials: that they would learn, using the computer, the methodology of problem solving in contract law. This explanation lasted only a few minutes. I took the students through the first few 'pages' of the CAL tutorials and then the students were on their own.

In the first year (82/83) I tended to remain in the tutorial room observing the students and the computers to make sure that the students could

cope; by and large they did. I answered no substantive questions during this time, being content to tell the students to type in the answer and see what the computer made of their answer. As the CAL tutorials progressed - and being only fortnightly about 4 CAL tutorials were covered by each student - I remained less on the scene so that for the last two tutorials I would turn up with the tutorial disks (and the disks on which to record the students' answers), make sure all was working and then leave, only returning about 50 minutes later at the end of the tutorial in order to collect all the disks.

In years two and three (83/84 and 84/85) following my introduction-to-computers talk and having taken them through the first few 'pages' of the first CAL tutorial, I left the students completely on their own. All I did was to appear at the beginning of each tutorial to hand out the disks, making sure everything was working and then leave. About 50 minutes later I would reappear to collect the disks.

(ii) Experimental design

I think that it is most important to incorporate CAL tuition into course as an actual part of the course. The traditional form of social scientific psychological experiment which takes the form of a carefull controlled experiment under laboratory conditions, is not, I think, appropriate for most teachers. This is because, whilst this is a well accepted form of experimentation, all it ever proves is that under those carefully controlled conditions the experiment either works or does not. In contrast to the normally carefully controlled laboratory conditions my 'experiments' were conducted as the normal part of the teaching course. No 'interested' volunterrs were asked for from amongst the students. They were assigned either to myself or my colleague according to timetable demands. Those students allocated to me became the CAL group of students and those students allocated to my colleague became the NONCAL group of students. The students had no choice in the matter. Those students designated CAL students received CAL tuition as part of their normal timetabled course. They received no extra tuition. The CAL tuition was instead of, not additional to, normal tuition. I think that this pragmatic approach to the introduction of CAL is what most teachers and Heads of Department really want to see because it gives them feedback as to whether CAL, if effective in the context of the course and clasroom.

It is interesting here to compare the way in which my CAL was used in comparison with the wat in which some CAL is used in the USA. My CAL was a component of the normal tuition; in the USA Maggs and Morgan (1975) observe:

> "... we [do] not deprive any student of live instruction in order to test the value of pure PLATO instruction. We could make the PLATO instruction an added benefit to some student, but we could not force students in a true "test group".

Although, then, the experimental arrangement used to test and evaluate the CAL tutorials was not the normal scientific arrangement[1] the actual results achieved by the method adopted reflected the use of CAL in the actual teaching environment. Such results are normally difficult to find in other studies because those studies usually go no further than testing a small discreet topic under strictly laboratory conditions and do not give as realistic results.[2] Notwithstanding the lack of laboratory conditions the CAL results have been compared with the NONCAL control group so as to give the scientific balance normally expected of such experimentation. On the whole, I consider that any disadvantages which arise out of this method of testing are more than outweighed by the real results achieved.

7. Student Feedback and Head's Reactions

In the first year, once the students had received their introductory talk on CAL and computers from me I stayed to observe and things started well. Observation showed that the students seemed keen and seemed to be working well with the computers. However, after the first few weeks some members of one particular tutorial group became unhappy with the tutorials because they wished to be able to talk to me, their tutor, instead of proceeding with their CAL tutorials.[3] I explained to the students that the CAL tutorials were soon to finish in any case, and as they were happy with this assurance, the CAL tutorials progressed. This 'unhappiness' reached the ears of the Head of Department. His reaction to this was apprehension in that he feared that if a student who had received CAL tuition failed his first-year examination he might use this as a ground for appeal against his failure. He suggested that the CAL tutorials be stopped and that conventional tutorials should re-commence. My reaction to this was that having analysed the students marks for their essays there was no academic reason to stop the CAL tutorials since the results showed that on average the CAL students had done slightly better than the NONCAL students. As a compromise I suggested that since the students were only receiving fortnightly tutorials they would only receive about 4 CAL tutorials (which amounted to about two thirds of the first terms tutorials). If it was perceived that the CAL students were behind the conventionally taught students at the end of the period of CAL tuition then the CAL students would be given weekly conventional tutorials (as opposed to fortnightly tutorials) from the beginning of term 2 until they had caught up with the conventionally taught students. This proved unnecessary. With these observations and undertakings the Head of Department was happy and the CAL tutorials progressed as planned.

One interesting incident of relevance here is that in the second year of our degree course I took the same group of students for Commercial Law who has done the CAL tutorials in the first year. The very group (and person) who in the first year had been responsible for trying to bring the CAL tutorials to an end asked if they could do CAL tutorials in Commercial Law!

8. Monitoring

Although it was not possible to make a hard copy of the students' answers whilst the students did their CAL tutorials, because of the absence of printers over the three years, all the students' answers to every question over the three years were recorded. All recorded student answers were printed out after each tutorial so that the students' answers could be compared with the model answers. If from these student answers it was discovered that the students were giving correct answers but the computer was rejecting them as incorrect because of the programming or oversight the programs were altered before the next group of students used the tutorials. In fact, very few parts of the program were actually altered for this reason.

Besides being able to keep check on the suitability of the CAL programs this monitoring process was also important because it enabled me to check on the progress of the students to see if they were encountering areas of difficulty which I would not have expected. The students' answers showed that although they did experience some areas of difficulty, these areas were ones that I had anticipated and these areas were the same ones that students experienced in conventional tutorials. This monitoring proved very helpful when the CAL tutorials needed defending (see 7. above).

POLITICS AND IMPLEMENTATION: CONCLUSIONS

As I mentioned earlier there are two main 'political' constraints to be dealt with when trying to introduce CAL into a department. Firstly, gaining the approval of the Head of Department. Secondly, gaining the approval of the other members of staff within the department.

The difficulty of implementing change is not exclusive to CAL. An interesting example is of the 'T.V. revolution' in teaching. Blumenfeld et al (1979) said

> "Educational television was originally perceived as a way to permit the 'master teacher to lecture to more students than was previously possible through live sessions. Has television been used to do what previously had been impossible or very difficult to accomplish in a regular classroom with traditional equipment? Typical uses of television indivate that established practices and assumptions about teaching have thwarted the diffusion of this technology (Kroeber, 1963, p.223). Innovation based on computer technology could suffer the same fate." p.187.

> "If school personnal [departmental members of staff - my words] do not participate in the innovation plan, it is unlikely that they will continue with CBE after the innovator leaves."

WHAT IF CAL 'GOES WRONG'?

Finally, once a teacher has decided to design, write and implement their own CAL tutorials they should not be put off from seeing their project through to the end. They need a fall-back position in case the unexpected happens but should not be pressured into abandoning the project. If properly thought out, planned, designed and implemented the introduction of CAL tutorials into a course should not prove too difficult and should prove most worthwhile.

POST SCRIPT

Despite the colleague with whom I shared the Contract Law course "coming around" to the use of the CAL tutorials, event to the extent of commenting

> "In my opinion the tutorials have served a very useful purpose in that they provide the student with an ability to approach contractual problems in a constructive and logical manner. They allow the student to break down a particular hypothetical question and avoid the inevitable problem of that student 'jumping in with both feet' in an effort to advise a particular person. The CAL tutorials also require the students to quote authority for any answer that is given which is invaluable to their future studies on the degree."

> "This year (year 5 86/87) we intend giving all students at least 3 CAL tutorials, and it is our hope and belief that they will provide the student with a solid basis on which to tackle contractual issues of a problematical nature."

After 5 years of using the CAL tutorials I was seconded out of the Law Department for a year. Because I was not around to book the computer room, give out the disks etc. the CAL tutorials were not used in the sixth, and subsequently in the seventh year. Next year, however, the Law Department will have its own computing facility and although I will probably not be teaching Contract Law again I intend to put the CAL tutorials up on the computers so students will have easy access to them even if they are not formally used in the Contract Law course.

---

(1)The same difficulties of arranging a sound scientific experiment has been experienced in the USA: see Park in The CALI Report.Vol.3,No.1,p.3.

(2) Dobson C.B.et al in Understanding Psychology (1981) Weidenfeld and Nicholson at p.6 say

> "It has been estimated that nearly 90% of psychology experiments use American college or university students as subjects."

(3) See Keller F.S.and his comments on students' need for reassurance in "Good-Bye, Teacher...", Journal of Applied Behaviour Analysis AEEE Part 1, 78.

ARTIFICIAL INTELLIGENCE: WHAT EDUCATORS NEED TO KNOW

G.F. Cartwright, L. Coffin
Dept. of Educational Psychology and Counselling
McGill University
Montreal, PQ

Researchers agree that the potential of artificial intelligence (AI) is high. Though artificial intelligence research has spanned several decades, progress in the area is now accelerating. General applications of artificial intelligence techniques are increasingly available in commercial software, and specific applications in education, particularly in logic programming, are becoming more common.

Short term predictions are that an increasing number of programs will become available which claim to use artificial intelligence techniques. While some of these programs will live up to these claims, others may not. Of major concern is the lack of familiarity among educators with the field of artificial intelligence.

This paper suggests that educators need to understand the basic principles of AI and be aware of their educational applications in order to evaluate critically software which purports to be artificially intelligent.

INTELLIGENCE ARTIFICIELLE : CE QUE LES ÉDUCATEURS DOIVENT SAVOIR

Les chercheurs reconnaissent les grandes possibilités de l'intelligence artificielle (IA). La recherche en ce domaine s'est étalée sur plusieurs décennies; cependant, c'est maintenant que les progrès en ce domaine se font de plus en plus grands. Des applications générales des techniques de l'intelligence artificielle envahissent de plus en plus le secteur commercial et des applications en éducation, notamment en programmation logique, sont plus courantes.

Des prévisions à court terme permettent de croire qu'on disposera d'un nombre de plus en plus grand de programmes faisant appel aux techniques de l'intelligence artificielle. Certains de ces programmes se révéleront efficaces et d'autres peut-être pas. Ce qui doit surtout nous préoccuper, c'est le manque de connaissances des éducateurs dans le domaine de l'intelligence artificielle.

Dans le présent document, on estime que les éducateurs doivent comprendre les principes fondamentaux de l'intelligence artificielle et en connaître les applications éducatives pour être en mesure d'évaluer, avec un esprit critique, les logiciels que l'on dit artificiellement intelligents.

ARTIFICIAL INTELLIGENCE: WHAT EDUCATORS NEED TO KNOW

G. Cartwright and L. Coffin

Department of Educational Psychology and Counselling
McGill University
Montreal, Quebec

Artificial intelligence (AI) techniques attempt to produce in computers behavior similar to intelligent behavior in humans (Dede, 1986; Jackson, 1985; Stevens, 1985). Applied to instruction, it is possible to imagine intelligent computer-assisted instruction (ICAI) at one end of a computer-aided learning (CAL) continuum, with traditional computer-assisted instruction (CAI) at the other. Most modern instructional systems fall somewhere along this continuum.

The CAL Continuum

EDUCATIONAL TREND ——>

CAI <----------> ICAI

Figure 1

Traditional CAI centers on providing a technological framework to transmit an educator's pedagogical decisions. ICAI, using artificial intelligence techniques, shifts the emphasis toward capturing the knowledge necessary in the first place for experts to make instructional decisions. Once captured, programs can be developed which dynamically compose instructional dialogs (Wenger, 1987). The trend in education is steadily shifting toward the ICAI end of the continuum (Dede & Swigger, 1988).

The primary reasons for this shift include:

1. the recent advances in artificial intelligence theory and software, coupled with
2. a growing understanding by educators that instructional systems must contain dynamic models of the student, the task, and the instructional discourse (Clancey, 1986; Keller, 1987), and the realization that traditional, frame-based CAI cannot achieve these ends.

Additional indirect support for this shift comes from:

3. the increasing performance/price ratio of microcomputers;
4. the increasing number of personal computers;
5. the increasing familiarity with computers among educators; and
6. the heightened general expectancies for computer performance.

The Challenge for Educators

The challenge for educators is to face this trend towards more ICAI in education, to understand its implications, and to explore its applications in instructional settings.

Unfortunately, educators lack sufficient theoretical and practical knowledge of AI. This contributes to the difficulty of understanding ICAI, of applying it in the classroom, and of dealing with the expected dramatic increase in the availability of commercial software which claims, accurately or inaccurately, to be AI-based.

At the same time, educators have become overly concerned with the *content* of software rather than the *process*. More and more educational software packages allow teachers to modify content. This is especially so in the case of expert systems which permit the use of entirely new knowledge bases. Because undesirable content can be easily modified, teachers now need to shift their attention to the pedagogical process underlying the instruction.

A fundamental difference between ICAI and CAI is that ICAI is based on an information processing approach. The programs do not simply relay information, but attempt to determine the structure of this information, understand what students are doing, and diagnose problems (Waldrop, 1987). Consequently, educators must step back from the content and look more closely at the techniques.

For example, they need to ask, "Does the program learn?" One such early program was called *Animals*. It was written in Logo and attempted to guess what animal a student was thinking of by asking a series of questions. If the program guessed incorrectly, the student was asked to

provide information about the chosen animal. The program absorbed this information, built up its knowledge base, and could then use the new information the next time the program was executed. In effect, the program learned.

"Does the program attempt to diagnose student errors?" An early example of this type of program was called *Debuggy* and was developed by Burton (1982). *Debuggy* was designed to diagnose student errors in subtraction problems. It analyzed a student's responses to problems, narrowing the possible errors a student could make in order to diagnose a specific error.

"Does the program use students' own input and enter into a dialogue with them about their work?" Keller (1982) developed a program called COMCON (Comma Conversar) which, rather than pick a sentence from an existing bank of questions to teach the correct use of the comma, allowed students to compose and enter their own sentences. The program then proceeded to converse with the student about the correctness of their sentence and through the course of the conversation built a model to represent the student's abilities (Keller, 1987).

While such programs may seem fairly simple, they represent the first steps on the road to ICAI. What differentiates these programs from traditional CAI are their underlying instructional strategies. Indeed, a good, intelligent tutoring system should employ instructional strategies indistinguishable from those of a good, human tutor.

What Educators Need to Know

AI is likely to impinge on all aspects of life. For this reason, it is crucial that educators, more than anyone, understand the nature of AI and its implications for education.

AI holds the potential for strengthening the person/tool relationship, increasing the human intelligence/creativity component, emphasizing the recognition and solving of unusual problems, de-emphasizing basic skill fluency, and re-emphasizing conceptual mastery, all necessitating a shift in the conventional educational paradigm. According to Dede (1988):

> *Person/tool partnerships require more worker intelligence than present occupations, since the human cognitive strengths in the synergy involve skills such as creativity, flexibility, decision making given incomplete data, complex pattern recognition, information evaluation/synthesis, and holistic thinking. Such higher-order mental attributes might become a new definition of human intelligence, as basic cognitive skills would increasingly shift to the AI tool's portion of the partnership* (p. 125).

Educators need to view AI, not as a threat, but as a useful tool in both teaching and understanding the nature of human learning and intelligence. This they can learn from AI researchers who, in their attempts to create machines which simulate human intelligence, must necessarily study what it means to be intelligent. Their insights and knowledge about how humans think can enable educators to better understand how learning occurs. This kind of understanding holds significant implications for the development of more effective educational materials and instructional strategies (Brady, 1986).

Educators need to understand the relationship between AI and knowledge mastery. Specifically, they need to appreciate that ICAI shifts the emphasis away from teachers as subject matter specialists to the computer. Teachers, according to Romiszowski (1987):

> *. . . may no longer need to be 'subject matter experts' in the way they are today, but will be very much more the facilitators of learning, the socializing force, the good example to be followed and many other such 'roles'. . . A big change, perhaps, but a change in the direction of 'high-touch'* (p. 102).

Educators need to be comfortable with this new role.

They also need to explore possible applications of AI in their classrooms. They need to understand the role of intelligent databases, expert systems, and intelligent tutoring systems. They need to be able to use them themselves and to instruct their students on the use of these tools.

Finally, educators need to be able to assess the claims of software manufacturers to distinguish which programs are truly AI-based. Already, teachers have become critical of what passes for software innovation. They now know that a program is not necessarily intelligent just because it refers to students by name, or remembers where they left off, or keeps track of their progress, though it may have seemed so to them a decade ago.

Where Do We Go From Here?

We are at a turning point. While the computer revolution in the schools is over insofar as convincing educators of the need for acquiring hardware, it is just beginning with respect to convincing them of the need for new and more powerful, intelligent software. It is now necessary to begin the large-scale training of both pre-service and in-service teachers in the new AI-based software. Educators need to be provided with training and support if they are to be adequately prepared for their changing role and able to face the challenge AI presents to education.

References

Clancey, W. J. (1986). Qualitative student models. *Annual Review of Computer Science. 1*, 381-450.

Brady, H. (1986). Artificial intelligence: What's in it for educators. *Classroom Computer Learning*, *6*(4), 26-29.

Burton, R.R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman and J. Brown, (Eds.), *Intelligent Tutoring Systems*. New York: Academic Press.

Dede, C.J. (1986). A review and synthesis of recent research in ICAI. *International Journal of Man-Machine studies*, *24*, 329-253.

Dede, C.J. (1988). Probable evolution of artificial-intelligence-based educational devices. *Technological Forecasting and Social Change*, *34*, 115-133.

Dede, C.J., & Swigger, K.S. (1988). The Evolution of instructional design principles for intelligent computer-assisted instruction. *Journal of Instructional Development*, *11*(1), 15-22.

Jackson, P.C. (1985). *Introduction to Artificial Intelligence*, (2nd ed.). New York: Dover Publications Inc.

Keller, A. (1982). *The Comma Converser*. Unpublished Ph.D. dissertation. Montreal QC: Concordia University.

Keller, A. (1987). *When Machines Teach*. New York: Harper and Row.

Romiszowski, A.J. (1987). Artificial intelligence and expert systems in education: potential promise or threat to teachers? *Educational Media International*, *24*(2), 96-104.

Sleeman, D., and Brown, J.S. (Eds.), (1982). *Intelligent Tutoring Systems*. New York: Academic Press.

Stevens, L. (1985). *Artificial Intelligence: The Search for the Perfect Machine*. New Jersey: Hayden Book Company.

Waldrop, M. M. (1987). *Man-Made Minds*. New York: Walker and Company.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. CA: Los Altos. Morgan Kaufmann Publishers, Inc.

TEACHER ADAPTATION TO COMPUTERS IN THE CLASSROOM

B. Durell
Faculty of Education
University of Toronto
Toronto, ON

Theory suggests that the introduction of computers into classrooms should be preceded by extensive training of teachers in the use of the machines and the courseware that the machines deliver. The needs of the classroom should first be determined, then software to meet these needs should be selected or produced and finally machines to run the software should be provided. In reality, computers are often introduced into schools for a variety of reasons, some of which are obscure or poorly thought out. Therefore teachers may find themselves with computers in their classrooms although no, or little, teacher training has taken place and the courseware available has not been selected to meet particular needs. We have had the opportunity to study a JK-8 suburban school into which the school board has placed a minimum of three ICON computers for each classroom. This situation differs from the usual study of computers in schools in which researchers bring computers to schools, conduct intensive teacher training and give extensive support to teachers during the implementation. In the school under study, the school board has provided the computers, training and teacher support. It is their project which we are studying. The researchers do not have a commitment to the success of the implementation. We are conducting qualitative research into this implementation process. One of the most interesting findings to date has been the ways in which teachers adapt to the presence of the computers. The findings have implications for teacher preparation, courseware design, implementation of technology in schools and development of teacher expertise.

ADAPTATION DES PROFESSEURS AUX ORDINATEURS DANS LES SALLES DE COURS

En théorie, avant d'introduire des ordinateurs dans les salles de cours, on devrait toujours donner une formation poussée aux professeurs dans l'utilisation des machines et des didacticiels que ces dernières permettent d'exploiter. Il faudrait déterminer alors les besoins de la classe, puis sélectionner ou produire le logiciel permettant de satisfaire à ces besoins et enfin fournir les machines permettant d'exploiter le logiciel. En réalité, les ordinateurs entrent souvent dans les écoles pour diverses raisons dont certaines sont obscures ou mal déterminées. Il peut donc arriver que des professeurs aient dans leur classe des ordinateurs, qu'ils soient peu ou mal préparés et que l'on n'ait pas choisi le didacticiel permettant de répondre aux besoins particuliers du groupe. Nous avons eu la possibilité d'étudier une école de banlieu (JK-8) où la commission scolaire a doté chaque classe d'au moins trois ordinateurs ICON. Cette situation diffère de ce qui se passe normalement dans les écoles où les chercheurs introduisent les ordinateurs dans les écoles, assurent une formation poussée des professeurs et offrent un soutien important aux professeurs pendant la période d'implantation. Dans l'école que nous avons étudiée, la commission scolaire a fourni les ordinateurs, la formation et le soutien aux professeurs. C'est leur projet que nous sommes en train d'étudier. Les chercheurs n'ont pas d'intérêt dans la réussite de l'implantation. Nous effectuons une recherche qualitative sur de ce processus d'implantation. L'une des découvertes les plus intéressantes jusqu'ici, c'est la façon dont les professeurs s'adaptent à la présence des ordinateurs. Les conclusions ont des incidences pour ce qui est de la préparation du professeur, la conception des didacticiels, l'implantation de la technologie dans les écoles et le perfectionnement des professeurs.

TEACHER ADAPTATION TO COMPUTERS IN THE CLASSROOM

Brian Durell

Faculty of Education
University of Toronto

Most discussions of using computers in schools are based on a number of quite sensible assumptions. It is assumed that the needs of the teachers and students will be assessed. Once needs are known, procedures for meeting the needs will be selected. Where those procedures indicate that computers could be used effectively, appropriate existing software packages are selected or new software is developed. Computer hardware that can run the necessary software is acquired and installed. Finally, it is assumed that teachers and students will be given thorough and systematic introductions to the use of the hardware and software.

During the last two and a half years, my colleagues and I have studied the introduction of computers into a JK-8 school. Most of the common assumptions about how computers should be brought into schools were violated, for legitimate reasons, in this implementation. Rather than producing chaos this situation has revealed facets of the implementation process that would not come to light in a more structured study.

THE SETTING

A suburban school board chose a JK-8 school as the site for a pilot project in computers in the classroom. ICON educational microcomputers were installed. The UNISYS ICON is one of the Grant Eligible Microcomputer Systems approved by the Ontario Ministry of Education. All classrooms are equipped with at least three computers and a printer. Some have as many as six ICONs. School offices and seminar rooms also have computers installed.

There are 142 ICON-2s in the school, served by 11 fileservers, and 41 printers one of which is a colour printer. The fileservers are networked so that students may log in on any computer in the school and have access to their personal files.

Our research team was invited to investigate the implementation. This is an important differnce from many other studies of computers in school. This project is the Board's. We have no vested interest in its success. We study how the computers are used.

Some initial training was given to the teachers, but they have not had detailed exposure to all of the software available to them and their students. Nor were they given a prescription for how to use the computers in their classrooms. They were encouraged to use them, but not required to do so.

Except for the high level of computer equipment in the school, this site offers a more typical example of bringing computers into schools than would be found in most research projects.

METHODOLOGY

We have adopted a qualitative approach (Jacobs, 1987; Lincoln & Cuba, 1985) to studying the use of computers in the school. Five researchers observe within classrooms. Each observer visits two classrooms for a half day per week per classroom. Observation notes are entered into a text base for later analysis. Publications of the school related to the computer project, samples of student work, and out-of-class comments of teachers are other sources of data. Interviews and questionnaires are administered to both teachers and students.

The research team provides formative feedback to school and Board staff, equipment manufacturer's representatives and the Ministry of Education through a project steering committee.

FINDINGS - TEACHER ADAPTATION

A significant finding of our research has been the ways in which teachers have adapted to the computers in their classrooms. Adaptation is a dynamic process. Adaptation will be examined in terms of four dimensions - psychological, physical, curriculum integration, and scheduling student use.

The teaching staff have expressed a wide range of attitudes toward the computers. The attitudes form a rough scale: enthusiastic -- interested-- cautious -- opposed. A teacher's placement on the attitude scale gives little indication of how that teacher adapts to the computers in the classroom.

Psychological - Hall (1974, 1979) strives to understand adoption of innovations by attending to the adopter, not the innovation itself. How adopters perceive an innovation, their attitudes and feelings toward the innovation are crucial in determining whether an innovation will, in fact, be adopted.

The teachers studied in this project showed concerns about the computers which were thrust into their classrooms. Many of them were defensive about the machines. They were quick to proclaim their ignorance of how the computers worked and of how to use them in their classrooms. This position could provide protection from responsibility for how the computers were used.

Computers were made available to some teachers to take home over the summer. This opportunity for using the computers at their own pace and without students, researchers, and other teachers looking over their shoulders, produced major changes in the way these teachers related to the computers.

In the second project year a computer committee was formed. Eight teachers comprise the committee. They organize information and idea-sharing sessions for the school staff and produce a newsletter about computer use in the school. Teachers tend to be more relaxed about having the computer in their classrooms and are investigating the available software with an eye to using it in their teaching.

Physical - The physical dimension of teacher adaptation to the computers is shown in where and how they place machines in the classroom, and the degree to which the teachers interact with the students who are using the computers.

In the first year many computers were banished to back corners of classrooms. It would be a mistake to read too much into the placement of the computers alone. However, machines placed in a back corner of the classroom and a teacher who never goes near students who are using the computers suggest a teacher who has safely isolated the threatening machines from the regular flow of classroom activities while ostensibly providing the expected computer experience for students.

In the second year several classrooms had four, five or six computers. There was a tendency for these computers to become a focal element of the classroom. In some cases the computers occupy a complete wall of the room rather than just a corner.

The willingness of teachers to approach the computers to monitor student work or to give help with problems is not clearly related to teachers' attitudes about the computers. A few enthusiastic teachers never approach the computers in use. Some cautious teachers regularly check the work of students at the machines and give what assistance they can when problems arise.

Integration into curriculum - Initially, almost all of the teachers provided computer time for their students, but gave little direction about what the students should do at the computers. The machines served as electronic smorgasbords. Toward the end of the first year some teachers began to have students produce assignments in curricular areas using word processing and desktop publishing programs. This practice is is much more common in the second year. Primary level teachers also make use of arithmetic drill programs in a systematic manner.

Again, teacher attitudes do not predict the degree to which the computers are used for curriculum-related activities.

Allocation of student time - Almost all of the teachers set up schedules of times when students would use the computers. Most tried to give every student some computer time each day or on at least four days out of five. The priority of computer use varies from teacher to teacher. In some classrooms the periods of computer usage are quite restricted. In most, computer time covers most of each school day, but students may be called away from the machines to attend lessons that the teacher considers particularly important. In a few classrooms, computer use has the highest priority. Students are responsible for catching up on any topics whose presentation they miss due to using the computers, but no presentation preempts computer time in these classrooms.

Teacher attitude and relationship of computer activities to the curriculum are not related in any simple way to the priority given to student computer time.

CONCLUSION

Our findings suggest that the introduction of prospective teachers to computers in education should be conducted at a leisurely pace. Teachers need time to explore the machine and determine how they relate to it at a personal level.

Technological innovations in schools should not be presented in teacher-proof packages. Teachers should be encouraged to express their concerns and reservations. They should be given the chance to work out their own ways of relating to and making use of any innovation. Teachers know the classroom world and often will generate better ways of using innovative materials and devices, if given the chance, than might be suggested by outside experts.

A teacher who has developed expertise in classroom operation may be put into the uncomfortable position of performing as a novice when computers enter the classroom. Teachers need time to develop to an expert level of computer use (Dreyfus & Dreyfus, 1981; Berliner, 1986).

Teachers are intelligent professionals. They do not have to be told how to do things, although they appreciate assistance in unfamiliar areas. The teachers in this study have been innovative in working out ways to make use of the computing power put into their classrooms. They have blended the power of the computers with the needs of their students and their own personal teaching styles to produce a wide range of effective ways of using computers in their classrooms.

REFERENCES

1. Berliner, D.C. (1986). In pursuit of the expert pedagogue. Educational Researcher, 15(7), 5-13.

2. Dreyfus, H.L., & Dreyfus; S.E. (1981). Putting computers in their proper place: Analysis versus intuition in the classroom. Teachers College Record, 85(4), 578-601.

3. Hall, G.E. (1974). The concerns-based adoption model: A developmental conceptualization of the adoption process within educational institutions. Austin: University of Texas, Research and Development Center for Teacher Education. (ERIC Document Reproduction Service No. ED 111 791).

4. Hall, G.E. (1979). The concerns-based approach to facilitating change. Educational Horizons, 54(1), 202-208.

5. Jacob, E. (Spring 1987) Qualitative research traditions: A review. Review of Educational Research, 57(1), 1-50.

6. Lincoln, Y., & Guba, E. (1985). Naturalistic inquiry. Beverly Hills, CA: Sage.

IMPLEMENTATION OF CAL MODULE FOR THE TEACHING OF CHEMISTRY

S. Al-Yamani
Arabian Gulf University
Manama, Bahrain

One of the most important factors in sixth form (A-level) teaching is the variation in the assimilation rate of the students. The time taken to understand a given body of information will vary considerably from one student to another but, more than this, a student's ability to assimilate information varies with the nature of that information.

A second factor which can pose problems is the number of students in sixth form classes, and a third important factor applies to first year degree courses, as students from different schools have different levels and amounts of knowledge.

This paper concerns itself with the use of a CAL module for the teaching of organic nomenclature to sixth form and first year undergraduate organic chemistry students. The software was tested on a number of students, typical of those for whom the software was intended, and then was revised according to the comments provided by the students and the teachers. The overall results obtained from these field-tests confirmed the effectiveness of CAL in teaching, and the high standard of the software.

MISE EN OEUVRE DU MODULE EAO POUR L'ENSEIGNEMENT DE LA CHIMIE

L'un des éléments importants de l'enseignement dans les classes de première (niveau A) est la différence de vitesse d'assimilation des étudiants. Le temps nécessaire à la compréhension d'une matière varie considérablement d'un étudiant à un autre et, en plus, la capacité d'assimilation d'un étudiant varie également selon la nature des données transmises.

Un deuxième facteur important susceptible d'entraîner des problèmes est le nombre d'étudiants des classes de première. Le troisième facteur touche les cours donnés dans les classes de sixième car le niveau et la quantité de connaissances des étudiants provenant d'écoles différentes peuvent varier considérablement.

Le présent document traite de l'utilisation de l'EAO pour l'enseignement de la nomenclature organique aux étudiants des classes de première et aux étudiants de première année du baccalauréat en chimie organique. On a fait tester le logiciel par un certain nombre d'étudiants qui répondaient parfaitement au profil de l'étudiant cible. Il a ensuite été corrigé en fonction des commentaires des étudiants et des enseignants. Les résultats globaux de ces essais en milieu opérationnel ont confirmé l'efficacité de l'EAO dans le domaine de l'enseignement et la qualité supérieure du logiciel.

IMPLEMENTATION OF CAL MODULE FOR THE TEACHING OF CHEMISTRY

Dr. S. Al-Yamani

Education Department, Arabian Gulf University, Bahrain

One of the important factors in sixth form (A-level) teaching is the variation in the assimilation rate of the students. The time taken to understand a given body of information will vary considerably from one student to another but, more than this, a student's ability to assimilate information varies with the nature of that information.

A second improtant factor which can pose problems is the number of students in sixth form classes, and a third important factor is the different levels and amounts of knowledge of students in first year university degree courses.

This paper concerns the use of CAL module for the teaching of Organic Nomenclature to sixth form and first year undergraduate organic chemistry students. It was thought that the CAL technique might be able to overcome all the problems mentioned above, mainly because: (i) different students can complete a common cours in different periods of time, and will have been working at their own pace throughout; (ii) large numbers of students can study as individuals; and (iii) by using CAL the lecturer could attempt to ensure that all students begin the course with the same foundation.

DEVELOPING THE CAL MODULE

Three major stages were followed during the development of the CAL module (Fig. 1).

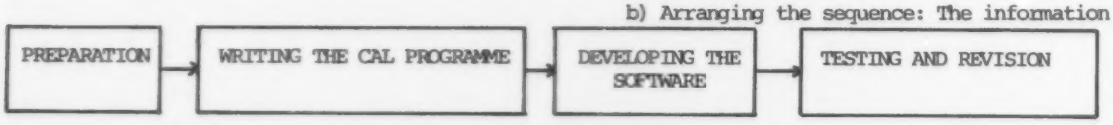


Fig. 1. Major stages in the development of the CAL Module.

1. PREPARATION

At this stage the following were defined:

a) The standard of learners: Sixth form and first year undergraduate organic chemistry students.

b) The study topic: Organic Nomenclature - Alkanes.

c) The objectives of the CAL programme: These are stated in precise and measurable terms in the guidance sheet which can be found in the booklet accompanying the software.

d) The pre-requisite knowledge: The student is assumed to have passed O-level chemistry or its equivalent.

2. WRITING THE CAL PROGRAMME

There are several steps at this stage: Writing the pre-test, arranging the sequence, writing the frames, and writing the post-test.

a) The writing of the pre-test: The purposes of this test were the following:

(i) To give the students an idea of the kind of problems they will be able to solve when they finish the programme.

(ii) To find out if the students need to go through the programme or not. If students achieve a minimum score (75%), then they do not need to work through the programme.

(iii) To enable the teacher to find out whether the students have achieved the programme's objectives or not (by comparing their scores in the pre-test and in the post-test).

(iv) To assist in the writing of the programme.

b) Arranging the sequence: The information was organised into a logical teaching sequence suited to the students' ability level and based on Skinner's principles, i.e. deciding the objectives of the programme; analysing these into responses; and teaching these responses one by one (Leith, 1966).

c) The writing of the frames: The material for the entire programme was first written as for normal prose writing, with particular reference to Johnson (1976). At a later stage this was

converted to normal programme presentation (i.e. frames, questions and answers), which allows the material to be worked on only one frame at a time. These frames were examined very carefully in order to avoid the common mistake of attempting to teach too much in too short a time, and were related to the pre-test and the objectives. In addition, each rule was illustrated by a series of questions and answers.

Since frames in a linear programme are small, it was decided to use what is called a panel (Figs. 2 & 3), containing more extended material. Each panel presents one new idea and the most important point is made clear and obvious in order to avoid any confusion. Various methods, including several different examples, brackets, arrows, upper-case letters, and sub-headings were used to focus on the important point which the panel is presenting.

The first ten straight-chain alkanes are

NAME	No. OF CARBON ATOMS IN CHAIN	STRUCTURE
Methane	1	CH_4
Ethane	2	CH_3CH_3
Propane	3	$CH_3CH_2CH_3$
Butane	4	$CH_3CH_2CH_2CH_3$
Pentane	5	$CH_3(CH_2)_3CH_3$
Hexane	6	$CH_3(CH_2)_4CH_3$
Heptane	7	$CH_3(CH_2)_5CH_3$
Octane	8	$CH_3(CH_2)_6CH_3$
Nonane	9	$CH_3(CH_2)_7CH_3$
Decane	10	$CH_3(CH_2)_8CH_3$

Note: the general formula is C_nH_{2n+2}

Fig. 2

The carbons that are not included in the parent chain are considered as substituents and are named as alkyl groups.

To obtain the name of an alkyl group, the -ane of the alkane name is re-placed with -yl.

For example :

methane : CH_4

methyl : CH_3-

propane : $CH_3\,CH_2\,CH_3$

propyl : $CH_3\,CH_2\,CH_2$-

Fig. 3

These panels were usually followed by a series of questions and answers. The type of questions used were multiple choice questions and constructed response questions. Multiple choice questions (Fig. 4), take more time to write than other types of objective questions, but in the author's experience students respond favourably to this type of question. To discourage guessing and prevent students from assuming that they understand a specific concept because of a chance response, the following procedures were used:

(i) some of these questions have an "I don't know" or "Help" options;

(ii) four option items were used most of the time;

(iii) all answers were plausible; and

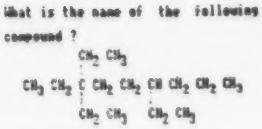
(iv) a variety of questions and examples were employed.

Constructed response questions (Fig. 5), however, have to be concise and clear and the answers required have to be sufficiently defined to allow a tight marking scheme. Prompts were used in gradually reduced frequency to help the students.

This is designed to encourage students to depend more upon themselves and to think more creatively.

QUESTION

What is the name of the following compound ?



1- 6,3,3-triethylnonane

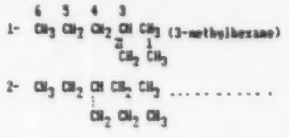
2- 3,3,6-triethylnonane .

3- 4,7,7-triethylnonane .

Fig. 4

QUESTION

Name the second compound :



1- $CH_3\,CH_2\,CH_2\,CH\,CH_3$ (3-methylhexane)

2- $CH_3\,CH_2\,CH\,CH_2\,CH_3$ .....

Fig. 5

If the students chose or wrote the correct answer to a question, the frame to which they were sent contained the correct response confirming their answer and a new question was given. If the students chose or wrote an incorrect answer, the frame they saw presented remedial information, and directed them: (i) to the previous frame to try again; or (ii) to a review statement; or (iii) to a new question similar to the one they had just answered.

In addition, loops were used so that students could rapidly pass through material that had already been mastered, but receive extra information and tuition on those aspects of the topic on which performance was still weak.

After a series of questions and answers, a review statement (Fig. 6), was provided to summarise the sequence of the frames which had just been learned. The review statement was intended to focus on the concepts being taught and to help the students to tie their new ideas together, and to help ensure a mastery of learning.

REVIEW

The presence of identical substituents is indicated by an appropriate multiplying prefix :

di for two

tri for three

tetra for four , ect.

Fig. 6

Therefore, the general sequence of frames used in the programme consisted of the following elements:

PANEL - QUESTIONS & ANSWERS - REVIEW

In addition, at the end of the programme a summary was given for revision purposes, in order to focus on the important rules and concepts which were learned throughout the programme.

d) The writing of the post-test: The post-test was constructed to be the same format as the pre-test but using different questions. This ensured objective evaluation of what was learned from the programme, while preventing the effect of learning from the pre-test.

3. DEVELOPING THE SOFTWARE

Before developing the software the programme was checked and read critically to ensure that it was of the required standard.

In spite of the fact that the programme could be produced in a printed format as a booklet, a microcomputer was chosen for the presentation of the materials of the programme for the following reasons:

(i) computers are widely available in schools and universities;

(ii) it allows complete control of the students' rate of working through the programme;

(iii) it provides an immediate feedback to students;

(iv) it provides students' scores in the pre-test and in the post-test;

(v) it prevents cheating;

(vi) the programme can be used by one student or simultaneously by a number of students;

(vii) students can easily work on the computer, therefore the time taken to adjust to the method of working will be minimal; and

(viii) it provides clear presentation of the subject matter which can be enhanced by illustrations using computer graphics.

Selection of a Microcomputer:

The selection of a particular computer for developing this CAL programme was influenced by the following factors:

a) Availability and easy access: The computer had to be available at a large number of schools and universities so that students, who ideally should be familiar with the system, could have easy access to the computer unit whenever required.

b) Presentation facilities: The computer had to be capable of presenting pages of text which can be mixed with graphics at a reasonable resolution (for chemical symbols, tables, illustrations, etc.), in order to present the subject matter clearly.

c) Network facilities: The facility of networking several computers together was needed, so that they can share essential input/output (I/O) devices such as printers, disk drives, etc., and also allow students to work simultaneously.

d) Memory requirement: The computer had to have a reasonable amount of memory in order to satisfy the programmer's requirements.

The computer which satisfied the above criteria was the BBC model B microcomputer.

Software Design and Implementation:

The programme is mainly composed of information presented in the form of text and chemical structural formulae. The main problem encountered in designing the computer program was the display of both text and chemical symbols at the same time, thus the use of a Graphics Mode was inevitable. Owing to the memory limitation, a compromise between the resolution of the graphics mode and the amount of memory available for the program was necessary. Therefore, Mode 4 was used as the main display mode consuming only 10 Kbytes of screen memory and providing a reasonable graphics resolution.

TEXT:

In order to minimise the size of the computer prog am due to the limitation imposed by the available memory (RAM), the text required for each frame was stored in a single random-access file on disk. Text was initially prepared using a word processor in order to provide a clear presentation of justified text, thus eliminating the problem of having incomplete words at the end of lines. The text for each frame was written separately and marked in the file by means of an identification number which, offset from the beginning of the file, was calculated and stored in a talbe. A BASIC sub-routine was written to load the relevant part of the text pertaining to a particular frame by reading its offset from the table. Then the text file was opened and the file pointer moved to the appropriate point in the file before the text was read.

CHEMICAL STRUCTURAL FORMULAE:

The presentation of chemical structural formulae on the screen required the use of subscripts for the numbers in chemical compounds. The basic principles of the procedure used to do this was the use of a joint text and graphics cursor. Similar routines were written to draw vertical, horizontal, and diagonal bonds between atoms. In order to enable the students to enter their answers after each question, two routines were written which are foolproof and user-friendly. The first deals with constructed response questions, which accepts both upper and lower case characters and ignors spaces. The second deals with multiple choice questions, which ignores any numbers/characters typed outside the given range.

THE SOFTWARE PACKAGE:

The entire program was divided into manageable sub-programs which were chained together in a manner which is transparent to the user. The suit of programs was stored on a floppy disk (40 or 80 tracks) and an automatic boot facility was included in order to facilitate the operation of the program for the user. The booting procedure is started by pressing the SHIFT and BREAK keys simultaneously. This loads the BASIC file BOOT which starts by displaying the title page (screen dump technique), followed by a menu which prompts the user to choose which part of the CAL programme is required from the following files:

(i) First time user: Allows the students to go through the programme in the normal logical sequence.

(ii) TEST 0: Pre-test.

(iii) MAIN 0: First part of the teaching programme.

(iv) MAIN 1: Second part of the teaching programme.

(v) MAIN 2: Third part of the teaching programme.

(vi) TEST 1: Post-test.

These files were added on the basis of teachers' requests on a questionnaire evaluation, so that students can work in different sessions without the need for repetition.

The instructions needed to use the CAL programme (software), were written on a guidance sheet, which contained the following: (i) the aim of the programme; (ii) background about organic nomenclature; (iii) instructions on how to use the programme; (iv) the objectives of the CAL programme; and (v) how to run the programme.

In addition, a summary sheet intended for student reference, was normally handed to the students during their working sessions.

4. TESTING AND REVISION

Testing & revision is one of the most important features of CAL technique. The software has to go through many revisions before practical use.

The software was used in two distinct ways in the evaluation. The first test of the software was with first year A-level students in their third term, with experience of the subject, who used the programme as a revision exercise (14 students-survey 1). Feedback was obtained through evaluation questionnaires and consequently the software was altered on the basis of the students' and teachers' comments. Owing to the good result obtained from the above questionnaires and the students' scores in the post-test, a decision was made to use it for providing basic teaching.

Therefore, the second test of the software was with new entrants to A-level courses in their first term, with no experience of the subject, who used the programme as a teaching exercise (4 students-survey 2). This was again followed by revisions based on the feedback obtained from evaluation questionnaires.

Fig. 7 states the scores from the second survey, which shows that all four students had been very successful in learning the CAL programme.

Pre-test scores	2	2	0	1
Post-test scores	14	15	14	14

Fig. 7. Comparison of scores on pre-test and post-test (scores obtained on both test out of 16)

EVALUATING THE SOFTWARE:

Three different methods of evaluation were used to evaluate the software. These were:

(a) The software evaluation schedule: This was designed by the author as a guide for writing the CAL programme. The purpose of this schedule was to ensure that a consistent, detailed and careful assessment of the software could be made.

(b) A questionnare of students who were using the software.

(c) An open-ended questionnaire survey of teachers of the classes using the software.

Possibly the most significant outcome of the evaluation of the software was the results of the questionnaires. Although the number of students who worked throughout the programme was small, the detailed nature of their responses provided useful comments for the evaluation purpose. Most of the students stated that they felt they had learned more easily by this method of instruction. Furthermore, most of them felt that this software was very interesting, easy to follow, had clear contents and a variety of examples. However, students were aware of the limitations of this method of instruction as regards to its impersonal nature. They felt the presence of the teacher is required to complement the effectiveness of this method.

CONCLUSION

In developing the software, a great deal of attention was paid to "foolproof" and "user-friendly" features with special emphasis on the quality of presentation. The software was designed in a modular fashion depending heavily on the use of sub-routines (procedures) to handle repetitive tasks. The overall results obtained from this study confirmed the usefulness of CAL in teaching in general and the effectiveness of this CAL module in teaching, or revising organic nomenclature.

REFERENCES:

1. Leith, G. O. M. The Writing of Programs. In Leith, G. O. M., Peel, E. A., and Curr, W. A Handbook of Programmed Learning. U.K. University of Birmingham, 1966.

2. Johnson, C. R. Organic Nomenclature: A Programmed Study Guide. U.S.A. Worth Publishers, 1976.

THE COURSEWARE DEVELOPMENT PROJECT: THREE YEARS ON

D.P. Sheridan
School of Business Administration
Dalhousie University
Halifax, NS

In 1985, the School of Business Administration, Dalhousie University, Halifax, Nova Scotia embarked upon a remarkable project to introduce a wide variety of computing devices and software directly into the curriculum. Initially an office of the future environment was established with the help of IBM PCs and compatibles for every faculty member and a substantial student laboratory. Each device was connected via a digital switch to three resident, Ethernetted MicroVAX IIs, and to the university's mainframe. A computer-managed learning environment was created using CML from CBTS, Calgary and a front-end was built using a CAL language called Producer. Micro-based CAL was developed in several languages and is now in daily use on the School's LAN. Corporate data from several billion dollar companies is incorporated into the curriculum via a 4 GL and expert system applications have been developed which involve a large crown corporation. This paper will report on the success of implementing CAL and CML in the graduate and undergraduate programs, the development environment (staff and funding needed for such an undertaking), and the overall effects on a faculty who had, at the outset, ten dumb terminals and three Apple IIe's. The School of Business is now considered one of the most technologically advanced in North America and has its software running in many campuses in Canada, USA and Australia.

LE PROJET DE DÉVELOPPEMENT DE DIDACTICIELS: UN PROGRAMME DE TROIS ANS

En 1985, l'École des hautes études commerciales de l'Université Dalhousie, à Halifax, en Nouvelle-Écosse, s'est engagée dans un projet remarquable visant à introduire directement dans le programme d'études une vaste gamme d'appareils informatiques et de logiciels. On a d'abord établi un bureau de l'avenir en fournissant des IBM PC et des compatibles à tous les professeurs et un laboratoire bien équipé aux étudiants. Chaque appareil était relié, via un communicateur numérique, à trois MicroVAX II résidents reliés à un réseau Ethernett et à l'ordinateur central de l'université. On a créé un environnement d'enseignement assisté par ordinateur à l'aide du CML de CBTS, Calgary, et l'on a construit un frontal en utilisant un langage EAO appelé Producer. L'enseignement assisté par micro-ordinateur a été développé dans plusieurs langages et il est maintenant utilisé quotidiennement dans le réseau local de l'École. Des données provenant de plusieurs sociétés ayant un chiffre d'affaires de l'ordre du millard de dollars sont incorporées au programme d'étude via un 4GL, et des applications de systèmes experts ont été développées avec la collaboration d'une société de détail importante. Cette communication expliquera le succès de l'EAO et du CNL dans les programmes des premier et deuxième cycles, le cadre de développement (le personnel et le financement nécessaires à une telle entreprise) et les effets généraux sur une faculté qui, au départ, avait dix terminaux "non intelligents" et trois Apple II. L'École des hautes études commerciales est maintenant considérée comme l'une des plus avancées en Amérique du Nord sur le plan technologique et ses logiciels sont utilisés dans beaucoup de campus au Canada, aux États-Unis et en Australie.

THE COURSEWARE DEVELOPMENT PROJECT: THREE YEARS ON

D.P. Sheridan
School of Business Administration
Dalhousie University
Halifax, NS

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

ADULT LITERACY AND TECHNOLOGY: AN OTTAWA BOARD OF EDUCATION STOREFRONT PROJECT

S. Nutbrown Massey
The Learning Centre
Ottawa Board of Education
Ottawa, ON

The Learning Centre, a technology-for-literacy centre administered by the Ottawa Board of Education, is the first storefront adult literacy program in Canada to integrate computer technology into the learning program. The Centre is a partnership between industry, government, educational institutions and community. A three-year bilingual project, the Centre has four objectives: 1) to provide direct service to adults in need of basic skills improvement by establishing an educational model incoporating technology; 2) to provide training for teachers, administrators, tutors, graduate students and cooperative education students from the adult and secondary-school panels; 3) to evaluate commercially available software and generate our own software which enhances adult learning and promotes maximum learner control and flexibility, and 4) to research the value and applicability of computers in basic literacy instruction.

This paper will present an update on Centre operations, particularly with regard to the use of technology in the learning programs, in courseware development and in administration; an overview of our professional development strategy; a listing and evaluation of commercially available software and a description (and perhaps demonstration) of the coursewa e developed in-house; and the results or status of our research into the role of computer technology in basic literacy instruction. Because we are in the unique role of serving as a model for other literacy programs entering the computer age, we will conclude by addressing some of the problems and issues in the integration of computer technology in adult literacy programs.

ALPHABÉTISATION DES ADULTES ET TECHNOLOGIE : UN PROJET DU CONSEIL SCOLAIRE D'OTTAWA

Le Centre de lecture et d'écriture, centre qui met la technologie au service de l'alphabétisation et qui est administré par le Conseil scolaire d'Ottawa, est le premier programme public d'alphabétisation des adultes au Canada à utiliser la technologie informatique dans le cadre d'un programme d'enseignement. Le Centre est né d'une association entre le secteur industriel, le gouvernement, les établissements d'enseignement et la communauté. Ce projet bilingue d'une durée de trois ans doit atteindre les objectifs suivants: 1) dispenser des services directs aux adultes qui ont besoin de parfaire leurs connaissances de base en instaurant un programme d'enseignement assisté par ordinateur; 2) former les enseignants, les administrateurs, les directeurs d'études et les étudiants en enseignement coopératif provenant des groupes de l'enseignement aux adultes et du niveau secondaire; 3) évaluer les logiciels commerciaux sur le marché et produire ses propres logiciels dans le but d'améliorer l'apprentissage des adultes et d'offrir une meilleure supervision des étudiants et une souplesse accrue; et 4) déterminer la valeur et l'applicabilité des ordinateurs dans le cadre de l'alphabétisation.

Ce document offre une mise à jour des opérations du Centre, surtout en ce qui a trait à l'utilisation de la technologie au sein des programmes d'apprentissage, à la mise au point de didacticiels et à l'administration, un aperçu général de la stratégie de perfectionnement professionnel, une liste et une évaluation des logiciels commerciaux sur le marché et une description (et éventuellement une démonstration) du didacticiel maison, les résultats ou l'état de nos recherches sur le rôle de l'informatique dans le domaine de l'alphabétisation. Comme notre unique rôle est d'ouvrir la voie aux autres programmes d'alphabétisation à l'ère de l'ordinateur, nous terminerons par l'étude des problèmes et des questions que soulève l'intégration de l'informatique dans les programmes d'alphabétisation des adultes.

Adult Literacy and Technology: An Ottawa Board of Education Storefront Project

Susan Nutbrown Massey

The Learning Centre
Ottawa Board of Education

INTRODUCTION

Because of its economic and social repercussions, functional literacy is an international concern. In the Ottawa-Carleton region alone, 23% or one in five adults cannot read, write, or handle numbers well enough to function in today's society. In Canada, the statistics range from 17% in British Columbia to 44% in Newfoundland.

The Learning Centre, a storefront program, was established to help resolve the problem of functional illiteracy. The Learning Centre is a three-year bilingual participatory research project which incorporates computer technology with the latest approaches in adult education to provide basic skill instruction. In addition to literacy and numeracy skills, computers can provide learners with marketable skills in our technological world. Computer-assisted instruction can also provide adults with an opportunity to see this technology used in programs as well as to use this technology themselves.

PARTNERS

The Learning Centre was set up as an example of "Partners in Education." It is administered by the Ottawa Board of Education and supported by the following founding partners: Apple Canada, Apple Canada Education Foundation, Carleton University, Ottawa-Carleton Learning Foundation, and Secretary of State.

After the project approval, other organizations came on line as our supporting partners: Autoskill, Inc., Computer Connection, MECC (Software) Inc., Microsoft Canada, Ottawa-Carleton Coalition for Literacy, Ontario Ministry of Skills Development, Ontario Ministry of Education, Rideau Street Merchants Association, Robinson/Ogilvy Ltd., Sony of Canada Ltd., The Ottawa Citizen, and UNISYS.

OBJECTIVES

The major objectives of The Learning Centre are:

. To provide direct service to adults in need of basic skills improvement by establishing an educational model incorporating technology.
. To provide training for teachers, administrators, tutors, graduate students and cooperative education students from the adult and secondary-school panels.
. To evaluate commercially available software and generate our own software which enhances adult learning and promotes maximum learner control and flexibility.
. To research the value and applicability of computers in basic literacy instruction.

STAFF

Eleven people work at the Centre:

. Centre Coordinator (full-time)
. Curriculum/Courseware Developer (part-time)
. Intake Coordinator (part-time)
. Research Coordinator (part-time)
. Five instructors (part-time)
. Two secretary/receptionists (one full-time and one part-time)

Three instructors teach thirty hours a week; two teach fifteen hours each. The Coordinator, Curriculum/Courseware Developer, and the full-time Secretary/Receptionist work thirty hours a week.

HOURS

The Centre is open forty-five hours a week:

Monday and Tuesday	9:00 - 4:30
Wednesday and Thursday	9:00 - 8:30
Friday	12:30 - 4:30
Saturday	1:30 - 4:30

INTAKE/ASSESSMENT/ORIENTATION

Each learner undergoes an intake procedure that gives as much information as possible about the learner's strengths and weaknesses for the learner, for the instructors, and for research purposes.

A Learning Plan is completed after the learner has been in the Centre for about two or three weeks. At this time, the instructor, in collaboration with the Intake Coordinator and the learner, completes the form. General and specific objectives are set, as well as suggestions for evaluating progress toward attainment of these objectives. The plan is both formal and informal: formal in terms of asking for a commitment on the part of the learner to set a weekly schedule for attendance at the Centre, a schedule determined solely by our open hours and the learner's own demands on his time; informal in the sense that the objectives set and the suggested content and evaluation procedure are flexible. The goals are modified as the learner and the

instructor come to view more clearly the learner's strengths, weaknesses, and personal commitments.

LEARNERS AND INSTRUCTORS

Not all learners coming to the Centre must work on the computers. We have found that, after an initial shyness and perhaps trepidation, the learners are eager to work part of their time on the computers in the Centre. However, a computer is not meant to take the place of instructors or print materials. A courseware or software package is only as good as its instructional design. We are in the process of evaluating the software for its applicability to our learning situation. And, as teachers become familiar with more courseware packages and programs, the content they suggest and prepare for learners changes and becomes refined. The gaps we document between available material and student objectives will be filled by the software we develop at the Centre.

The instructors prepare material for each learner. This material, and the objectives they help the learner meet, is based on goals seen by the individual learner as practical and worthwhile. The adult learner in The Learning Centre participates fully in the development of his learning experience. S/he determines the schedule, the objectives the mode of delivery, and so on. We are also planning to have learner input into the software evaluation process.

PROJECTED OUTCOMES

The Centre's objectives are perhaps more concretely understood by our projected outcomes, of which there are eleven:

1. Provide information on incorporating computer technology into literacy programs. From the initial stages of planning and organizing the Centre to the actual use of computers with the learners, we are able to provide members of the educational community, as well as the general public, with information about the role of technology in adult literacy training. We are also able to provide information about the evolution of learner and instructor perceptions of the computer's role in adult basic education. We have been approached by people in the Northwest Territories: Yukon College; Camosun College in Victoria, B.C.; provincial education and training ministries from New Brunswick and Nova Scotia; and so on.

2. Develop software applicable to the learning needs of the functionally illiterate adult learner. Our first development projects involved administrative and operational uses of software and computer technology. We developed, tested, and revised forms that we would need in the day-to-day operation of the Centre. Some of these forms were developed in English only, some in French. We are in the process of standardizing the format of all forms and ensuring that forms are available in English and French.

As we previewed and evaluated commercially available software, it became evident that, even when a piece of software had excellent educational value, it often did not reflect an adult reality. In fact, most of the software that aims to develop basic letter and word recognition skills as well as basic reading skills is geared to children between four or five and eight. A presentation using cartoon characters or child-like sounds and noises does not respect the adult's dignity. Clearly, software with an adult context and clearly identifiable objectives would have to be developed. So we thought! However, our adult learners have outguessed us, not only accepting the childishness of some of the programs, but also thoroughly enjoying programs aimed at a child of seven.

In spite of the learners enjoying some of the software geared toward children, we are developing our own software that not only provides a realistic, Canadian context for the adult learner, but also is geared to the beginning literacy learner. We are also in the process of recording entry-level skills of literacy students with a view towards developing the accompanying software.

3. Provide training and materials to teachers who want to develop computer software in basic skill areas. The real training of our instructors began when they actually had to use the hardware and software in the preparation and implementation of the students' learning plans. Only then did instructors begin to experiment, to see the real implications of good courseware, to learn to combine in judicious doses print and computer activities, to be more discriminating in evaluating commercial software.

Our instructors had little, if any, experience in educational applications of computer technology, let alone the computer technology itself. Their anxiety and trepidation during the four weeks prior to the arrival of our first learners had almost completely dissipated by the end of our first full open day. Because of our experience with the pre-opening training session with our own instructors, we believe we are better able to understand the implications of training educators in the use of computers with learners as well as in the development of literacy courseware. We have since implemented a series of three-hour hands-on sessions for adult basic education instructors on using computer technology with students. These sessions are held every Monday night and they are given by our own instructors.

4. Provide a site outside school for testing software designed for functionally illiterate adults. We have received major software contributions from MECC (Software) Inc., Microsoft Canada, and Autoskill Inc. MECC (Minnesota Educational Computing Consortium) has a long-standing reputation as the company for educational software. This software has been developed in most subject areas, at all levels, and for

some cognitive-skill development. While MECC software in language arts and math is usually geared toward the adolescent, we are in the process of evaluating its applicability for the adult learner.

Microsoft Canada has generously provided us with Microsoft Works, an integrated package which has word-processing, database, spreadsheet, and communications capabilities. While this software has, as yet, not been used by our learners, its use in Centre administration has been invaluable. We are interested in evaluating its effectiveness and efficiency in keeping and managing the records of a Centre our size.

Autoskill Inc. has asked us to research the use of its Autoskill program, designed to diagnose and train reading-disabled adolescents, with adult learners. With the training provided by Autoskill, the program and backup print material, our instructors are able to test learners for reading disabilities and provide further training for them. Autoskill has also been used for developing reading skills in non-reading-disabled learners and so some of our learners are following Autoskill as a reading program.

5. Provide a network for sharing information on the existence of basic skills software and specific areas. When our software inventory and curriculum development projects are in a more advanced form, we will be able to disseminate them to any interested individuals, associations, or institutions. We are also investigating the entry-levels demonstrated by our learners and the skills or objectives they feel they need. We are planning to integrate learner objectives and courseware.

In addition to our curriculum development and inventory plans, we are also part of a provincial literacy network, a project initiated through George Brown College in Toronto with funding from Secretary of State and Ontario Ministry of Skills Development. Our students have been communicating with learners in Toronto, Sudbury, Kingston, as well as other sites in Ontario.

6. Evaluate software and hardware used at the Centre for its suitability for use with functionally illiterate adults. As mentioned above, our software analysis is ongoing.

7. Examine longitudinal achievement of adults using computer-assisted instruction. This outcome involves more research-oriented work and indeed is one of the longer-term goals of the Centre. A Research Coordinator began his duties in January and he is collecting and analysing data which will give us concrete answers to some very real concerns and questions on the use of technology in basic education programs.

8. Determine the areas of greatest need and greatest effectiveness in software when offering a total curriculum for the adult learner. This outcome is closely related to outcomes 2, 4, and 5. The first software we have developed is for objectives we believe are in greatest need and which have the widest applicability. We do not want to reinvent the wheel; if we believe commercial or other educational software is adequately fulfilling a need, we will not initially develop software to meet the same objectives. Part of our value as a technology-for-literacy centre will be one of judgment: judging when and why and if and how certain objectives can be met with educational software.

9. Determine the relationship between student achievement and use of computers in delivering instruction. Again, this is first a research concern which has direct implication for the style and type of delivery we offer our learners. We have certain intuitions and subjective observations about how well or how much better our learners are doing in reading, writing, and basic math since coming to the Centre. We instinctively feel that the more difficult-to-measure increase in self-esteem is very evident. However, it is the role of the Research Coordinator to quantify and qualify our observations. We expect that we will be able to disseminate initial research results within several months.

10. Provide information to ensure replication of the Centre. Through our reports, articles and conference presentations, we will be able to describe from conception to actualization the steps (sometimes one forward and two backwards) that were taken, and that are being taken, by everyone involved in the Centre. The Centre is a developmental project; we know that not only is our concern the quality of instruction to our learners, but also the duty to be a model for other centres and institutions wishing to join us in our endeavors; we learn through everything we have done and do.

Thus, we are in a way duplicating the path of our learners who come to us with needs and goals; the Centre also has goals. We help our learners formulate learning plans to meet these goals; the project proposal and subsequent refinements were plans for the Centre to meet its goals and objectives. Certain activities and content are applied to the learners to help them attain their objectives; the Centre is working within certain operational guidelines and with certain educational and research orientations that will result in the long-term attainment of the goals. And, as our learners can evaluate the content or goals which they have set for themselves, so too is the Centre constantly evaluating its progress toward the objectives which have been set.

11. Disseminate the results of the research to other literacy centres as well as make available any materials generated by the Centre. As mentioned above, we intend to disseminate information about the Centre: its operations, its mandate, its learners, its development, its research. We will also make available to

interested associations, institutions, and individuals any materials generated by the Centre.

CONCLUSION

Despite the short time that we have been open for learners, we are pleased with the quality of instruction they are receiving, with the use of computer technology in their overall learning objectives, with the interest shown in us by the public in general and the literacy community in particular. We currently have 140 registered students who are working on literacy and numeracy objectives, some with the intention of going on to credit high-school courses, others with the goal of improving the day-to-day quality of their lives. Student reactions to computer use have exceeded our expectations. And as a Centre we continue to evolve and learn and improve.

We are most grateful to all our partners who have helped us demonstrate that there is a way of bridging the gap between "the school" and "the street."

INCORPORATING TELEVISION/VIDEO AND COMPUTERS INTO A CHANGING SCHOOL ENVIRONMENT

D. Sharon, R. Marom
Planning and Development Research
TVOntario
Toronto, ON

This session will discuss the process of incorporating instructional technology into the elementary and secondary school system, focussing on how this process relates to simultaneous changes in other areas of the school environment.

Based on four case studies, we will present and discuss views of the role of technology, i.e. school TV/video and computers, reported by school board officials, principals and teachers. The case studies will examine efforts to connect the introduction of technology with new curriculum guidelines or new teaching approaches. They will also look at how school boards and schools go about providing resources and educating teachers. While the research will focus more extensively on school TV/video, school use of computers as creative or instructional tools will be considered as well.

The research reported here is the first of four studies planned to look at all levels of involvement with technology in teaching and learning. Later studies will look in detail at how teachers integrate TV/video into their lessons and at the roles television and video play in students' learning.

INTÉGRATION DE LA TÉLÉVISION/VIDÉO ET DES ORDINATEURS DANS UN MILIEU SCOLAIRE EN PLEINE ÉVOLUTION

Dans cet exposé, nous traiterons de l'intégration de la technologie d'enseignement aux systèmes scolaires primaire et secondaire, et en particulier des rapports qu'il y a entre cette intégration et les changements simultanés qui se produisent ailleurs dans le milieu de l'enseignement.

En nous appuyant sur quatre études de cas, nous présenterons et nous examinerons l'idée que se font les administrateurs, les directeurs et les enseignants de la commission scolaire du rôle que va jouer la technologie à l'école (c'est-à-dire télévision/vidéo et ordinateurs). Dans les études de cas, nous analyserons les efforts déployés pour intégrer la technologie aux nouveaux programmes-cadres ou aux nouvelles approches pédagogiques. Nous examinerons également la façon dont les commissions scolaires et les établissements d'enseignement ont affecté des ressources et formé les enseignants. Notre recherche portera davantage sur l'utilisation de la télévision et de la vidéo à l'école, mais nous étudierons également l'utilisation des ordinateurs comme outils créatifs ou d'enseignement.

L'étude dont il est question actuellement est la première de quatre études destinées à analyser sous tous les aspects, l'apport de la technologie à l'enseignement et à l'apprentissage. Les autres études permettront d'approfondir la façon dont les enseignants intègrent la télévision et la vidéo à leurs cours, et le rôle que joueront ces derniers dans l'apprentissage des étudiants.

INCORPORATING TELEVISION/VIDEO AND COMPUTERS INTO A CHANGING SCHOOL ENVIRONMENT

D. Sharon, R. Marom
Planning and Development Research
TVOntario
Toronto, ON

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

AN IMPLEMENTATION MODEL TO DEAL WITH KEY ISSUES OF COMPUTERS IN SCHOOLS

M. Szabo, G.H. Buck
Dept. of Educational Psychology
University of Alberta
Edmonton, AB

Only six percent of professional educators are users of computer technology. While the number of schools with computers is impressive, the amount of computer time available to students is insignificant. Some argue that computer technology has failed to deliver promised results and should be abandoned. This paper argues that 1) progress made to date is on a reasonable time schedule, and 2) there are significant implementation issues which have not yet been successfully addressed. Only when these issues are properly dealt with can we begin to prepare professional educators to capitalize on the benefits and avoid pitfalls of computers in instruction. The major issues to be dealt with include the change process, the technology itself, and the timing. Research in the literature and conducted by the author indicates that proper attention to the change process will result in proper utilization of computer technology with greater speed and fewer mistakes. Studies from a variety of fields, including education, make it clear that lasting change is more likely to occur when 1) the problem is clearly articulated, 2) a short-term leadership Task Force is assembled to solve the problem, 3) the Task Force is given extensive freedom to develop and carry out the objectives of the solution, 4) change is focused at the building level, 5) administrative support is clear and visible, and 6) long range planning is included in the solution.

This paper will present a major model for computer implementation based on the six factors above. In this model, three major goals taken on by the Task Force include 1) to model exemplary use of computer technology, 2) to assume a major role in training and assisting colleagues, and 3) to develop multi-year change-over plans. A brief history of the success of this model will be presented.

MODÈLE DE MISE EN OEUVRE TRAITANT DES QUESTIONS IMPORTANTES RELATIVES À LA PRÉSENCE D'ORDINATEURS DANS LES ÉCOLES

Six pour cent seulement des éducateurs de profession font appel à la technologie informatique. Bien qu'un nombre impressionnant d'écoles aient été dotées d'ordinateurs, le temps machine mis à la disposition des élèves est négligeable. Pour certains, la technologie informatique n'a pas répondu aux attentes et devrait être écartée. Le présent document soutient, d'une part, que les progrès réalisés l'ont été selon un calendrier raisonnable et, d'autre part, que certaines questions importante touchant à la mise en place n'ont pas encore été résolues. Ce n'est qu'une fois ces questions traitées correctement que nous pourrons former les éducateurs à tirer parti des ordinateurs et à en éviter les inconvénients. Les grandes questions à traiter comprennent le processus de changement, la technologie elle-même ainsi que le choix du moment approprié. Les recherches effectuées par l'auteur dans la littérature révèlent qu'une étude attentive du processus de changement favorise une meilleure utilisation de la technologie informatique, en accroît la vitesse et réduit le nombre des erreurs. Des études réalisées dans un grand nombre de domaines, y compris l'éducation, indiquent clairement que la permanence du changement repose en partie sur un certain nombre de facteurs. Le problème doit être clairement énoncé, un groupe de travail coiffé d'une direction à court terme doit être constitué pour résoudre le problème, ce groupe doit avoir toute latitude pour déterminer et atteindre les objectifs reliés à la solution, le changement doit être axé principalement sur l'étape de la création, l'administration doit assurer un soutien clair et visible et la solution doit comporter une planification à long terme.

Le présent document contient un modèle de mise en oeuvre de la technologie informatique qui tient compte des six facteurs susmentionnés. Dans ce modèle, le groupe de travail vise trois grands objectifs, soit: modéliser une utilisation exemplaire de la technologie informatique, occuper un rôle important dans la formation et l'aide aux collègues et mettre au point des programmes de transition pluriannuels. Un bref historique du succès de ce modèle est également présenté.

AN IMPLEMENTATION MODEL TO DEAL WITH KEY ISSUES OF COMPUTERS IN SCHOOLS

Michael Szabo, George H. Buck

Department of Educational Psychology
University of Alberta

OVERVIEW

The use or lack thereof of computers in schools has been criticized on a number of fronts for being too little, inappropriate or having too much resistance from educators. The first thesis of this presentation is that the most visible barriers to "widespread" and "appropriate" use of computers in education have largely been overcome (or will soon be in the near future). The second and major thesis is that the most serious barrier remains unchallenged and largely unnoticed. Stated differently, this barrier is how to overcome the inertia against change characteristic of almost all organizations, whether they be educational, government, or commercial. In short, how does one take something as new and different (innovative) as microprocessors and integrate them into every day commonplace usage in education?

BARRIERS WHICH HAVE FALLEN OR WILL SOON FALL

Although much has been written about problems which will have to be overcome, one about which there is widespread agreement is the *utility factor* of microcomputers. That is, how can they be applied to improve the lot of education? Bent (1981) has argued that computers have utility in areas which are central to the way education operates. First, education stands firmly grounded in the **written word**. With text and graphics processing applications, the computer has outstanding capabilities to deal with the written word. Educators strongly emphasize **skills of seeking out information** to be used for wise decision making and databases are ideal places to store and retrieve information quickly and easily. **Thinking** skills of problem solving and logic are stressed by educators and the computer is a powerful logic machine whose processes can be laid bare, inspected, analyzed and applied to human logic development. As a tool to create **educational materials**, the graphics processing applications coupled with high quality laser printers enable educators to create professional looking materials to enhance classroom learning. Virtually all important **numerical calculation** done today is done by computer, easing the computational burden and freeing the learning process to focus on mathematical conceptual development. Finally the computer has shown to be a powerful ally in providing **computer assisted instruction**. Microcomputers have clearly overcome the lack of utility barrier.

The second major barrier has been the *transparency of the computer*, that is the ease of use. We refer to this as the "paper and pencil" phenomenon; tools will not be used until their simplicity of use approaches that of the paper and pencil. Pull-down menus, pointing device input mechanisms and operations which are intuitively understandable by analogy or iconic representation are opening the doors to the "other 96%" who are not now making use of computer technology.

The third barrier is *adoption time*, that is the lag between development of something new and its widespread application. This lag has been estimated to be 20 years for high tech developments and 40-50 years for educational developments. If one assumes that computers in education represent a mixture of high tech and educational development, it can be argued that the time lag will be a minimum of 20 years and have an upper limit of 50 years. Given 1950 as the birth of computers (the first commercial model sold), we might expect

widespread application to occur sometime after 1970 and before 2000. The computer adoption time barrier has fallen or will soon fall.

A fourth barrier is *cost*. That too is dropping in magnitude (a 25% reduction per year in hardware) and importance as we begin to understand the costs of not moving into computer technology. We still have to understand and come to grips with ongoing non-capital costs such as software and people expertise necessary to operate microcomputers, but there are signs that the experience is beginning to generate wisdom.

BARRIERS TO BE RECOGNIZED AND OVERCOME

In our analysis, we incorporate the remaining barriers under the category of innovation or change. In studying why organizations are so stable, it becomes apparent that they acquire powerful mechanisms to resist change. While this penchant for stability protects from harmful or frivolous change, it also thwarts change which might be useful or beneficial. It is this barrier which we do not clearly recognize nor understand how to overcome: it is also the heart of the problem of widespread and appropriate use of computers in education. Several questions will be posed below to illuminate the process of change.

WHO SHOULD GUIDE AND CARRY OUT THE CHANGE?

Here the experience is quite clear. When change is carried out within an organization, there are usually one or more dedicated, enthusiastic individuals who strongly believe in the new direction. They are willing to take on an overload and in some cases subvert formal channels (Pinchot, 1985) to see their dream come true. In the legitimized world, such a group is called a **task force**. Members are selected for their colleagial respect, dedication, and willingness to work on a short term basis to bring about specified change. When the change is successful, the task force returns to its normal work function (which it never really left in the first place).

WHAT CONDITIONS ARE NECESSARY TO SUPPORT THE TASK FORCE TEAM?

The first and foremost condition is the freedom to use their collective knowledge and skills to guide, shape, and direct the designated change. Given a broad mandate (see below), the task force develops it own objectives to meet that mandate. Who is in a better position to determine the "appropriateness" of use of computers in the classroom than those who spend the majority of their time in classrooms? Stated differently, those innovations which are dictated in both form and function by curriculum developers, consultants, administrators or other influences are highly resisted and often modified and diluted beyond recognition (Goodlad, 1983a). Educators are bright, highly motivated self-starters, not automatons who carry out board decisions with military precision and obedience. Snyder & Anderson (1987) contend teachers must be involved in the decision making process.

A second condition is to acknowledge that computer technology will impact virtually all levels and functions of education. One implication is that massive retraining will be required for all educational personnel, not just for the "computer teacher". The task is massive and will not be accomplished solely by individuals operating alone or by a few consultants, no matter how dedicated they may be. In fact, there is a clear need for the task force to take on a specific team and leadership role. This special team is called a **leadership task force team.**

Several implications arise from this recognition. First, the leadership task force must operate as a team large enough to share the work load, but small enough and appropriately composed so as to be able to function quickly and effectively. A second implication is that a changeover to widespread use of computer technology will not be accomplished overnight. In fact, there is good reason to believe a multi-year changeover plan will be required (Goodlad, 1983b). Recall that (a) the computer revolution is being hailed as bigger than the industrial revolution and (b) the latter took, by all reasonable accounts, about 150 years to develop fully!

The third condition is prior visible support and commitment for the general mandate (goals) of appropriate and widespread use of computers from the senior and middle levels of administration. This involves an understanding of the expectations to be placed on the task force, a willingness to commit modest resources

to the success of the venture, and a willingness to personally back up the importance of the venture through training and other involvement. Sergiovanni (1984) made a good case for the active involvement of administrators.

The fourth condition is that members of the leadership task force team must be willing to acquire the necessary skills to provide the leadership. I say willing because it is not wise to select only those with excellent skills for the team. This course of action will reinforce the idea that computer technology is only for those who are highly trained or somehow special. Nor is it wise to create new bureaucratic positions to guide the change because the people hired to the new position (e.g., computer coordinator) are resented for the fact that they (1) do not teach regular full loads (i.e., work in the trenches), (2) have access to 'extra' resources and (3) are far too few in number to be of any widespread help when assistance is needed.

A major component of success of the leadership task force team is that they be well trained before they embark on their tasks. Experience with other innovations suggests that the training should, at the minimum, equip the team to: (1) be exemplar users of the technology, (2) training and assist colleagues in the use of the technology, and (3) develop long-range changeover plans to move from the present level of usage to a desired level as set by the team.

In the remainder of the conference presentation, we will sketch out some of the details of a plan to create, train, and provide support for leadership task force teams to guide their educational colleagues to appropriate and widespread use of computer technologies.

REFERENCES

Bent, D. H. (1983). Challenges for the '80s. Computer Data, February, 40-43.

Goodlad, J. I. (1983a). A study of schooling: Some findings and hypotheses. Phi Delta Kappan, 64(7), 465-470.

Goodlad, J. I. (1983b). A study of schooling: Some findings and hypotheses. Phi Delta Kappan, 64(8), 552-558.

Pinchot, G. (1985). Intrapreneuring. New York: Harper and Row.

Sergiovanni, T. J. (1984). Leadership and excellence in schooling. Educational Leadership, 41(5), 4-13.

Snyder, K. J., & Anderson, R. H. (1987). What principals can learn from corporate management. Principal, 66(4), 22-26.

STUDENT AND EDUCATOR ADAPTATION TO COMPUTER-ASSISTED LEARNING IN CANADA'S ARCTIC: THE EXPERIENCE IN THE KEEWATIN REGION, NORTHWEST TERRITORIES

H. Wolter
Arctic College
Rankin Inlet, NT

P. Fahy
Research and Development
Alberta Vocational Centre
Edmonton, AB

R. Taylor
Advanced Education
Rankin Inlet, NT

A Computer-Assisted Learning (CAL) project using the Control Data PLATO system has been underway in the seven communities of the Keewatin Region, Northwest Territories, since August, 1987. The project has made available PLATO basic skills and high school upgrading, selected higher education subjects (principally math and sciences) and various business software applications (spreadsheets, data bases and word processing packages) to the region's predominantly Inuit population.

This presentation will summarize project components and outcomes, particularly related to a) student and instructor adaptation to the technology; b) student learning; and c) effects of CAL on the adult education centres in the region.

ADAPTATION DES ÉLÈVES ET DES PROFESSEURS À L'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR DANS L'ARCTIQUE CANADIEN : EXPÉRIENCE DE LA RÉGION DE KEEWATIN, TERRITOIRES DU NORD-OUEST

Un projet d'enseignement assisté par ordinateur (EAO) mené avec le système PLATO de Control Data est actuellement en cours dans sept collectivités de la région de Keewatin, Territoires du Nord-Ouest, depuis août 1987. Grâce à ce projet, la population de la région, en grande majorité Inuit, a accès aux matières élémentaires de PLATO, ainsi qu'aux matières de l'enseignement secondaire et à des sujets choisis de l'enseignement post-secondaire (principalement mathématiques et sciences); en outre, elle peut apprendre avec ce système divers logiciels de gestion (tableurs, bases de données et traitement de texte).

Dans cet exposé, on résume les composantes et les résultats de ce projet, en les regroupant d'après a) l'acclimatation des élèves et des professeurs à cette technologie, b) l'apprentissage par les étudiants et c) les effets de l'EAO sur les centres de formation des adultes de la région.

STUDENT AND EDUCATOR ADAPTATION TO COMPUTER-ASSISTED LEARNING IN CANADA'S ARCTIC: THE EXPERIENCE IN THE KEEWATIN REGION, NORTHWEST TERRITORIES

H. Wolter
Arctic College, Rankin Inlet, NT

P. Fahy
Alberta Vocational Centre, Edmonton, AB

R. Taylor
Superintendent of Advanced Education, Rankin Inlet, NT

INTRODUCTION

This is a summary report of the first year of a two year Computer Assisted Learning (CAL) Project being carried out with Inuit adult students in the Keewatin Region of the Northwest Territories.

BACKGROUND

The Keewatin Region is located on the eastern edge of the Northwest Territories, and comprises seven hamlets, or communities. Four of these hamlets, Arviat (formerly Eskimo Point), Whale Cove, Rankin Inlet and Chesterfield Inlet are situated on the west coast of Hudson Bay, directly north of the province of Manitoba. Baker Lake is about 200 miles inland from Chesterfield Inlet; Coral Harbour is located on South Hampton Island, and Repulse Bay on the Melville Peninsular. These isolated Communities can be reached only by plane, except for a short period in the summer when they can also be reached by boat. These seven communities' populations range in size from 190 (Whale Cove) to nearly 1,500 (Rankin Inlet). The Inuit population of these communities ranges from 77% in Rankin Inlet to 99% in Chesterfield Inlet.

The first language in the region is Inuktitut, of which there are many dialects. The functional literacy level in the English language is reached by about 23% of the population. There has been some form of adult education in the Keewatin Region for more than fifty years, beginning at the Catholic Missions, where adults were taught to read the bible, and continuing with informal training by community organizations. The Northwest Territories Department of Education established adult education centres in three communities in the 1970s; by 1987, there was an adult education centre in each community in the region.

As the communities moved from a traditional land-based economy (hunting, fishing) to a more wage-based economy, a demand for different marketable skills emerged. Students who saw little value in education when they were in school as children and adolescents found that they needed to be at least functionally literate in English to be employable as adults in this economy. They needed specific training for some jobs, and the language of that training was usually English.

PROJECT OBJECTIVES

The objective of this project was to conduct a pilot project in Computer-Assisted Learning (CAL) that would allow:

a) adults who have not succeeded in traditional education programs to work independently at their own level and receive regular constructive feedback; and

b) adults requiring specialized training to get it in their community, improving their chances either of getting employment or of raising their level of employment.

PROJECT GOALS

1. To improve and expand, through the implementation of Computer-Assisted Learning (CAL), academic upgrading, job readiness and special skill training for adults in the Keewatin Region. To respond to the special needs of unemployed Inuit who were not reached in the past by traditional training programs and methods.

2. To provide specialized skills training not usually available in small communities to unemployed Inuit and other members of the community. (Special attention was to be paid to areas of training where local employment was available.)

FUNDING

The Canada Employment and Immigration Innovations Program provided funding in the

amount of $1.285 million over two years (1987 to 1989) for computer hardware, educational software, development of Inuktitut courseware, contract instructors, furniture, administrative support and external evaluation. The Department of Advanced Education, Government of the Northwest Territories, provided the equivalent dollar amount in person years, facilities and program operation and maintenance costs.

EQUIPMENT

Computer systems supplied by Control Data Canada were installed in the 7 Keewatin communities in August, 1987. The systems consisted of a Local Area Network (LAN) of either 5 or 8 computer terminal workstations, a 140 megabyte file server, printer, and a communications modem.

At the beginning of the second year (September, 1988), the number of computers was increased to 10 to 12 in each community.

The PLATO package consisted of Basic Skills, and High School Skills (GED) curricula. PLATO was used up to three hours daily by students as part of the adult basic education (ABE) program; business applications (chiefly databases, word processing and spreadsheets) were also used extensively.

ANTICIPATED RESULTS

The "anticipated results" in the original proposal for the Innovations Project were as follows:

a) Attracting and maintaining interest of a greater segment of the target population in educational programs.
b) Producing faster progress in academic training.
c) Providing job readiness skills.
d) Increasing chances of getting employment or improving level of employment.
e) The creation of a new more effective educational model for Inuit students which might be used through the Arctic.

FINDINGS

In year 1 (1987-88) of the project, the following findings were recorded:

1. Attracting and maintaining the interest of a greater segment of the target population.

The presence of CAL increased, sometimes dramatically, the numbers of students in programs, especially in evening programs, and in programs for employed persons and others who would not normally have been accommodated in regular full-time programming. Also, one community (Chesterfield Inlet), which had not been able to sustain interest in traditional programs in recent years, was successful in the first year of this project in attracting and maintaining students because, in the opinion of the adult educator, of the presence of CAL.

On a daily basis attendance rates were about the same as in previous years in the Keewatin; however, given the fact that, compared with Kitikmeot region students, the Keewatin programs were attracting more students and students with more varied backgrounds (as described below), the fact that there was no difference in attendance rates may be taken as an indication of the power of CAL to maintain students' interest. [The Kitikmeot is a region of the Arctic to the west and north of the Keewatin, comprising the communities of Cambridge Bay, Spence Bay, Coppermine, Gjoa Haven, Holman Island, and Pelly Bay. The Kitikmeot served as a control region in this study.]

Regarding client characteristics, the following differences were noted: Keewatin students were older, academically higher (as measured by the TABE and average levels of previous education), had more recent work history, and were more often parents and spouses compared with Kitikmeot students.

Overall, the adult educators reported increased interest in the Adult Education Centres in the communities in the Keewatin region, better student retention rates, and more registrations by employed persons in special work-related CAL courses, as evidence that CAL had helped achieve this objective.

2. Producing faster progress in academic training.

Because of its familiarity to students and staff, and its historically wide-spread use in the Northwest Territories, the Tests of Adult Basic Education (TABE) (1976 version) were used to assess the speed and magnitude of academic progress. The TABE tests skills in reading, math and language, with scores in Vocabulary and Comprehension, Computation Concepts and Problem Solving, and Spelling and Mechanics. There is no written composition component.

TABE results showed average grade equivalency gains of 0.63 and 0.92 grade levels after 3 and 6 months, respectively, for 26 Keewatin students for whom 3 sets of tests (pretest, posttest 1 and posttest 2) were available. A comparison of the pretest grade equivalency average (6.5) with the April,

1988, posttest grade equivalency average (7.9) showed an increase of 1.4 grade levels. Data from the previous year in the Keewatin region showed an average increase of 0.8 grade levels, and results from the Kitikmeot region for the 1987/88 year indicated an average gain of 1.2 grade levels. No correlation was found between time spent in the program and gains in grade equivalency level for either the Keewatin CAL group or the Kitikmeot controls.

A large majority of both the adult educators and students in the Keewatin region attested to the efficacy of CAL as a learning enhancer. Specifically, self-pacing, privacy of results, freedom of choice, and the availability of various specialized subjects (word processing, database, and spreadsheets, as well as advanced academic subjects such as math and science, and job search skills) were cited. Motivation was also increased by CAL, as was self-confidence in learning ability. Learning by CAL was perceived as prestigious and more effective by participants and the community at large.

Overall, the conclusion was reached that while CAL did not result in a major difference in rate or magnitude of learning gains as measured by TABE, it did contribute to significant increases in enrollment, enrollment of a wider variety of students, notably better success in communities where previous adult education programs had failed or had done poorly, improved retention of students, and a greatly improved self-image for participants, and a better image for adult education in the participating communities.

3. Providing job readiness skills

Students gained computer-literacy skills as a by-product of their use of CAL, and some students who pursued specialized training (word processing, database, spreadsheets and advanced topics such as higher math, physics and chemistry) also gained specific technical skills. Most users appeared to develop enhanced self-confidence and reduced reservations about computers as tools in daily life and employment. Many students reported their intention to pursue further technical training in the computer field.

Student attitudes were affected by the total adult learning experience, including but not limited to CAL: relations with supervisors, awareness of the job market, ability and willingness to seek employment, economic responsibility, family support, and cooperativeness were all rated higher on a pre/post measurement scale after completion of the program. Students and adult educators attributed some of these changes to experiences made possible by the presence of CAL.

4. Increasing chances of getting employment or improving level of employment

Acquisition of greater academic skills and credentials was regarded as prima facie evidence of greater employability. In addition, students' attitudes and plans were surveyed with a variety of instruments to show their specific intentions regarding employment: these proved to be very firmly oriented towards full-time jobs in the students' present communities, chiefly in trade and clerical positions. Expected starting salaries seemed to be within reasonable limits. As well, some adult educators recounted cases where employers who knew of the computer element in the adult education program contacted the Centre seeking computer-literate, trainable employees. There were numerous cases of employed persons who formerly would not have accessed the Centre enrolling in computer-based academic upgrading and other training related to their jobs, thus increasing their promotability and potential level of employment. [Long-term follow-up of students will indicate whether these initial trends persisted after the project.]

5. Creation of a new, more effective educational model for Inuit students which might be used throughout the Arctic

Interviews and questionnaires administered to students and staff indicated strong support for CAL both as a component of upgrading, and as a tool for enhancing computer-literacy, and for specialized applications (word processing, database and spreadsheets). Both adult educators and students appeared to adapt successfully to this innovation, though technical problems in some communities impeded early adoption to some extent. The adult educators reported adjustments in their role as instructors, chiefly due to a reduction in routine clerical, recordkeeping and instructional demands, and increased time and opportunities for individual and small group student interaction. In general, the difference was regarded by the educators more as a change in tasks and operations rather than as a true change in role.

As they became familiar with CAL's potential, both the adult educators and the students identified needs for courseware development. Topics for development included both supplements to existing materials, and new materials, especially in the areas of native language and culture. The adult educators requested that they be permitted to participate closely in the design and development of these materials. In the second

year of the project, development and pilot testing of these types of materials was planned.

In sum, this project showed some of both the problems and exciting potentials of CAL in the North. Where it was most successful, CAL and the availability of the related sophisticated hardware and telecommunications equipment produced notable attitudinal and behavioral results and effects. Where it was less successful causes were most often related to technical failure or unrealistic expectations. Throughout the project, the incidence of these latter declined. Overall, the project has come to be seen as a successful example of the use of technology to overcome barriers to access for adults learning in Canada's remote areas.

INFORMATION/AUTOMATION: COMPUTERS AND CONTRADICTION IN THE ELEMENTARY CLASSROOM

A.I. Griffith
Dept. of Measurement, Evaluation and Computer Applications
Ontario Institute for Studies in Education
Toronto, ON

The social applications of computer technology have the potential for both increasing and decreasing the knowledge and control of workers over their work process (Noble, 1984; Zuboff, 1988). This contradiction is, in part, a feature of computers -- machines which facilitate an orderly filing, retrieving and sorting of particular kinds of information. It also arises in the applications, or uses of computers in the work setting. This paper takes up the information/automation contradiction as it impinges on the work of teachers in the classroom. The paper draws on research conducted in an elementary school in Ontario where the Board of Education has placed a minimum of three computers in each classroom. The paper will bring into view how this contradiction is experienced and managed in classrooms as well as in the relation between the classroom, the school and the Board of Education.

INFORMATION/AUTOMATISATION: ORDINATEURS ET CONTRADICTION À L'ÉLÉMENTAIRE

Les applications sociales de la technologie informatique peuvent augmenter et diminuer les connaissances et le contrôle des travailleurs sur leur travail (Noble, 1984; Zuboff, 1988). Cette contradiction est, dans une certaine mesure, une caractéristique des ordinateurs, machines qui facilitent le rangement, l'extraction et le classement ordonnés de certains types d'informations. Cette contradiction surgit également dans les applications ou utilisations des ordinateurs dans le milieu de travail. Cette communication traite des conséquences de la contradiction information/automatisation sur le travail des enseignants en classe. Elle s'inspire de la recherche effectuée dans une école élémentaire en Ontario où le conseil scolaire a placé au moins trois ordinateurs dans chaque classe. La communication montrera comment cette contradiction est vécue et gérée dans les classes de même que dans la relation qui existe entre la classe, l'école et le conseil scolaire.

Information / Automation: Computers and Contradiction in the Elementary Classroom

Dr. Alison I. Griffith

Department of Measurement, Evaluation, and Computer Applications
Ontario Institute for Studies in Education
252 Bloor St. W., Toronto M5S 1V6

The introduction of computers into a work setting transforms the work of that setting dramatically: some jobs are lost; new ones are created; others are changed; the amount of information available to managers is increased; the ergonomics of the work place shift; and so on. The computer has become both the basis for, and the transformative icon of radical change in the organization of work in our society (Turkle, 1984; Roszak, 1986.) In schools, no less than in other work sites, the computer is beginning to reshape work.

In this paper, we will situate ourselves in the 'work organization' of elementary schooling. This perspective allows us to trace some of the social intersections that produce what we see as the interaction between students, teachers and computers in the classroom. In particular, I want to outline a conceptual framework through which we can begin to see the linkages between the work of the classroom, the Ministry-organized production of standardized educational software, the management of teachers's work within a bureaucratic institutional form, and the ongoing societal negotiation of the definition of education.

I want to address a different set of questions than are usually taken up in the area of instructional technology and the education system: questions of gender and class divisions within the "web of computing" (Kling and Scacchi, 1982)that is being constructed in the public schooling system. The questions will focus a research topic that illuminates but is not limited to the everyday work of teachers and students in the schooling system.

COMPUTERS AND THE LABOUR PROCESS

Zuboff (1988), Shaiken (1984), Noble (1984), and Cockburn (1985), among others, have documented the changes in the labour process that accompany the introduction of computer-oriented technology into such disparate sites as clerical work, nursing care, pulp and paper mills, banks, and the machine tool industry. Noble draws on Braverman's (1974) account of the "deskilling" of labour under capitalism in his research on the machine tool industry. He argues that research and development choices of particular kinds of technology have shifted the basis of production knowledge from the shop floor to the managerial/ technical levels of production. A similar theme is the basis of Zuboff's work as she describes the "automate/ informate" contradiction within computerized work sites. Zuboff argues that the contradiction of computerized work processes are that they: 1) reduce the control of the worker over their work process through a reorganization of the workers bodily relation to their work; and 2) make visible to the workers the managerial organization of work thus increasing their information about the organizational relations as a whole. In other words, workers now occupy the same conceptual terrain as management, flattening the hierarchical relations of the modern corporation.

Cockburn's studies bring into view the gender differentiation and inequality which is an integral aspect of the labour process. She argues that the changes brought about by computer technology and its application in the work place have not shifted the power relations between men and women. Rather, the reorganization of work through computerization has, if anything, drawn the lines between women's work and men's work more strongly and thus, maintains the subordination of women in society. Kraut, Dumais and Koch (1989) consider the mass of contradictory findings within the labour process literature to show the range of effects of computer oriented technology on workers in different jobs and locations in the labour market.

When we turn our attention to the literature on computers in the schooling process, we find numerous studies of the computer in the classroom and its "effect" in the classroom. Paul Olson (1983:181-182) has noted the contradictory viewpoints held within the educational literature on computers and technological change: For some, computers will open new avenues of creative exploration; for others, computers are the thin edge of the inhuman wedge.

The studies range from accounts of the student-computer interaction in classrooms (Brine, 1988; Carmichael, Burnett, Higginson, Moore and Pollard, 1985; Heap, 1987; Larter, Graganca, and Rukavina, 1987; Moore 1987; Murray, Lines and Spumont, 1986); to the impact of computers on the pedagogical work of teachers (Rose, 1986); to analyses of the variety of computer usage in schools serving students from different social classes (Becker, 1982, 1987); to the gen-

dered character of computer use and educational software (Hawkins, 1984; Harisim and Smith, 1987); to the "effect" of computers on cognitive development (Mandinach, E.B. and M.C.Linn, 1986; Kulik, Kulik and Bangert-Drowns, 1985); to studies of teachers' attitudes to computers (Lumsden and Norris, 1985; Reed, 1986). The literature on the use of computers in the classroom has been marked by a generalized approval of computers as a learning tool, as well as documenting a wide range of teacher and student responses -- excitement, innovation, apathy and fear (Ragsdale, 1988). John Olson (1988) suggests that the educational enthusiasm for computer use is grounded in the promise of artificial intelligence.

The work of Michael Apple (1988) is unique in this area of computer use in schools. He situates school-based computer use in the textually-organized and bureaucratic schooling system divided along lines of gender, class and race. Focusing on teacher's work, he locates the introduction of computer technology within the ongoing political struggle for control of the teaching process, and the intensification of teacher's work. Outside of Apple's analysis of changes in teachers work in the United States, there are no studies of the re-organization of teaching work consequent to the introduction into the school of computer-oriented instructional technology. The next section begins to spell out the parameters of such a study.

RESEARCHING TEACHER'S WORK

We will use the concept of "work" as one organizing frame. This concept has some limitations but will give us entry to the actual processes which occur in schooling. We begin with a conception of teaching as a work process embedded in a textually organized bureaucracy. The notion of work used here is not one limited to or bounded by the wage-labour relation. Rather, work is conceived of as requiring energy, time, resources and as having a structured rhythm in which a variety of people participate. In this sense, the work of teachers, or administrators is not reduced to some common denominator. Instead, we want to retain the differences between work processes (both paid and unpaid), preserving their uniqueness without losing the sense of structured and intentional action within the context of the education system.

Using this generous concept of work and situating ourselves, first, in the work process of the classroom, we find that the work organization of teaching is bound by a standardized curricular process through which teachers and students must proceed. In the classroom, this is experienced as limitations of time (Manicom, 1987). The conditions of teachers work are located in the intersection of the standardized and mandatory curriculum with the educational, social and economic resources of the students' families (Griffith and Smith, 1989a, 1989b). Thus, the inclusion of computer-oriented work by students into the pedagogical processes of the classroom is shaped by the actual work organization of the teacher within the conditions of work of the classroom and the school.

Already, our exploration leads us outside the walls of the classroom, into the administrative organization of a mandatory curriculum and outside the school itself into the students' families. The methodological approach which will allow us to research these intersecting threads is an "institutional ethnography". This method was developed by Dorothy E. Smith (1987) as a feminist method of inquiry. It begins in everyday experience, in the actual activities within which our social world is constructed. Unlike most other ethnographic methods, we do not remain in the immediacy of everyday life. Nor do we assume that everything there is to know about a particular topic is to be found completely in the lives of those affected by it. The institutional ethnography investigates a "section" of the social world as seen from the viewpoint of those whose work is, in various ways, involved in its production. As we map the social relations which inter-sect in the everyday world, we can begin to see the complex of relations which organize everyday experience. In an institutional ethnography, we move from one site to another, and from one "level" of organization to another. These variety of "levels" come into view if we consider the various "sites" of our research interest.

One such "level" is the socially organized processes of the classroom that are embedded in a gender-differentiated profession. Elementary schools are primarily staffed by women while the leadership in the school and in the Board of Education is primarily male. As a gendered work organization, teaching has something in common with other work sites in which women are numerically dominant.

Cockburn's (1985) studies of the clothing trade, of a mail order house and of a medical x-ray lab as well as Zuboff's (1988) accounts of the changes in clerical work in banks and insurance offices bring into view the processes which originate in male-dominated management processes. The introduction of computer technology in these hierarchically organized work settings facilitates increased control of the (primarily female) work force by the (primarily male) management. Campbell and Jackson (1989) are currently researching these processes as they affect the nursing and social work profession.

The insights of this research on the impact of technology on the work of women in the labour market are invaluable as they bring into view the technologically-driven changes in working conditions as a series of choices made by actual people in organizations structured to hold unequal relations of power -- a feature of the teaching process no less than nursing or social work. But it would be a mistake to remain within the similarities. Indeed, the differences are as illuminating as the similarities. For example, women in professional occupations have traditionally worked within a "discretionary gap" -- in the gap between the managerial organization of their work and

their front-line interaction with their clients/patients/students. In nursing and social work, the reorganization of work has increased management control of the professional work process and reduced the "discretionary gap". In teaching, however, the history of the struggle between the management of teachers work and teachers control of their work in the classroom has resulted in a much wider "discretionary gap" than is usual in women's professional work. Thus, the introduction of computers and computer-oriented work processes has been very slow in the teaching profession.

At the level of the school, computer-organized work processes have entered teaching work through the administratively-organized standardized curriculum. Classroom computers are no longer the novelty they were five years ago. Computers are brought into the school in a variety of ways: the Board of Education sets a goal of 3 computers per classroom or a computer lab in every school; parents groups buy computers for the school; administrative staff and teachers agree to use scarce school funds for a computer buy; and so on. The computers are the software, typically standardized educational software, some developed under the umbrella of the Ministry of Education. Classroom work processes are beginning to shift through the use of a variety of software packages with teachers correcting or marking work done on the computer screen or hardcopy. Individual teachers are still able to choose whether to incorporate computer-oriented projects in their classroom pedagogy, but the expectations of (particularly) middle class parents and most administrative staff is that computers will be available to students within the curricular process.

The availability of computers includes the availability of software constructed for educational use. In the United States, standardization of the curriculum has been the mechanism for both "deskilling" and intensifying teacher's work, shifting the locus of pedagogical control from the classroom to the education administration. The development of standardized computer software has increased the intensification and deskilling of teachers (Apple, 1988). In Canada, standardization is more advanced in Ontario than in some other provinces. Teachers, the majority of whom are not familiar with software construction, often rely on the Ministry of Education, and the computer consultants in their Board to make software available in the school. Teacher training on computers and software is minimal (Braganca, 1988), and teachers often must depend on other teachers at their school for information on the "how to" of computer use. The generalized favour with which computers are held view frames computers in the school, providing no arena for disagreement other than the dissatisfied rumblings of a few teachers who "don't like computers".

CONCLUDING REMARKS

If we begin in teacher's work and ask how it comes to have the character it does, we can trace the threads, the social relations, that intersect in the classroom. It locates us in the actual work organization of teachers in the classroom, and of administrators in the local school, the Board of Education and the Ministry of Education. It allows us to include the resources families can provide for individual students and the school in the conditions of teachers work. We can begin to trace the processes through which computer-oriented technology is shaping the work of teachers, administratively and pedagogically. One such process is the placing of education software in the standardized curricular processes of the school system, extending the administrative relevances of the Ministry and the Board within the classroom and reducing teachers control of the pedagogical process. It allows us to investigate the contradiction cited by Zuboff and to ask what aspects of teacher's work are being "automated" and what aspects provide for "informating". Until we begin to trace these processes, the particularities of the transformation of schooling and teachers work through instructional technology remains invisible.

BIBLIOGRAPHY

Apple, Michael (1988) *Teachers and Texts: A Political Economy of Class and Gender Relations in Education* London: Routledge and Kegan Paul.

Becker, Henry J. (1982) *National survey of school uses of microcomputers (Interim Reports 1 - 6)* Baltimore, Md: Johns Hopkins University.

Becker, H.J. (1987) "Using Computers for Instruction: The Results and Implications of a National Survey may Surprise You." *Byte 12* (2) 149-162, Feb.

Braganca, Ruth (1987) *Microcomputers in Grades Four to Six: Responses From Teachers, Pupils and Parents.* Toronto: Toronto Board of Education.

Braverman, Harry (1974) *Labour and Monopoly Capital: The Degradation of Work in the Twentieth Century* New York: Monthly Review Press.

Brine, John W. (1988) *Collaborative Publishing and Word Processing in the Primary Curriculum: Technology and Side Effects.* Unpublished Ph.D. Dissertation, University of Toronto.

Carmichael, H.W., Burnett, J.K., Higginson, W.C., Moore, B.G, and Pollard, P.J. (1985) *Computers, Children and Classrooms: A Multi-site Evaluation of the Creative Use of Microcomputers by Elementary School Children,* Toronto: The Ontario Ministry of Education.

Cockburn, Cynthia (1985) *Machinery of Dominance: Women, Men and Technical Know-how,* Boston: Northeastern University Press.

Griffith, Alison I. and Dorothy E. Smith (1989a) "What Did You Do in School Today? Women's Educational Work in the Family." *Perspectives in Social Problems* Spring.

Harisim, Linda and Smith, D.E. (1987) *Final Report on the Women Educator's Computer Conferencing Research Project,* Toronto: Federation of Women Teachers Associations of Ontario.

Hawkins, J. (1984) *Computers and Girls: Rethinking the Issues* Technical Report No. 24, New York: Bank Street College of Education, Centre for Children and Technology, April.

Heap, James. (1987) "Organizational Features of Collaborative Editing Activities at a Computer." Paper prepared for the American Educational Research Association, Washington, April.

Kling, R. and Scacchi, W. (1982) "The Web of Computing: Computer Technology as Social Organization." *Advances in Computers, 21,* 1- 90.

Kulik, J.A. and C.C.Kulik and R.L. Bangert-Drowns. (1985) "Effectiveness of Computer-Based Education in Elementary Schools" in Gutkin, T.B and Elwork, A (eds.) *Computers in Human Behaviour 1* (1) 59-74.

Larter, S., Braganca, R. and Rukavina, I. (1987) *Writing with Microcomputers in the Elementary Grades: Process, Roles, Attitudes and Products, Education and Technology Series,* Ministry of Education, Province of Ontario, Toronto: Queens Printer for Ontario.

Lumsden, D.B. and C.A. Norris (1985) "A Survey of Teacher Attitudes and Beliefs related to Educational Computing" *Computers in the Schools 2* (1) 53-59, Spring.

Mandinach, E.B. and M.C.Linn (1986) "The Cognitive Effects of Computer Learning Environments" *Journal of Educational Computing Research 2* (4) 411-427.

Manicom, Ann. (1987) *Constituting Class Relations: The Social Organization of Teachers' Work.* Unpublished doctoral dissertation, University of Toronto, Toronto, Ont.

Moore, Shawn. (1987) "Writer-Helper Collaboration during the Composition Phase of Computer Story Writing in Primary Classroom Settings." Paper prepared for the Sixth International Human Sciences Research Conference, Ottawa, May.

Murray, Jack, Lines E., Sprumont, B. (1986) *Word Processing in Elementary Schools: Seven Case Studies* Education and Technology Series, Ontario Ministry of Education: Toronto.

Noble, David (1984) *Forces of Production: A Social History of Industrial Automation* New York: Alfred A. Knopf.

Olson, C. Paul (1983) "Who Computes?" In David Livingstone and Contributors *Critical Pedagogy and Cultural Power* Toronto: Garamond Press, p. 170-205.

Olson, John (1988) *Schoolworlds/Microworlds: Computers and the Culture of the Classroom,* Toronto: Pergamon Press.

Ragsdale, Ronald G. (1988) *Permissible Computing in Education: Values, Assumptions and Needs* New York: Praeger.

Reed, W.M. (1986) "Teachers' Attitudes Toward Educational Computing: Instructional Uses, Misuses, and Needed Improvements." *Computers in the Schools 3* (2) 73-80, Summer.

Rose, G.A. (1986) *Understanding the Interaction between Teachers and Computers.* Unpublished doctoral dissertation, University of Toronto.

Roszak, Theodore (1986) *The Cult of Information: The Folklore of Computers and the True Art of Thinking* New York: Pantheon Books.

Shaiken, Harley (1984) *Work Transformed: Automation and Labour in the Computer Age* New York: Holt, Reinhart and Winston.

Smith, Dorothy E. (1987) *The Everyday World as Problematic: A Feminist Sociology* Boston: Northeastern University Press.

Smith, Dorothy E. and Alison I. Griffith (1989b) *Coordinating the Uncoordinated: How Mothers Manage the School Day.* Perspectives in Social Problems Spring.

Turkle, Sherry (1984) *The Second Self: Computers and the Human Spirit* New York: Simon and Schuster.

Zuboff, Shoshana (1988) *In the Age of the Smart Machine: The Future of Work and Power* New York: Basic Books.

INTERACTIVE VIDEO/VIDÉO INTERACTIF

SOFTWARE ENGINEERING TECHNIQUES FOR INTERACTIVE VIDEODISC APPLICATIONS

D. Bell, A.K. Anwar
School of Electronics
West Glamorgan Institute of Higher Education
Mount Pleasant, UK

P.G. Barker
School of Information Engineering
Teesside Polytechnic
Middlesbrough, UK

This paper describes the development of a portable environment for interactive videodisc applications. This environment is referred to, in this paper, as BIIVA, BASIC Interpreter for Interactive Videodisc Applications. There are many authoring languages available for courseware development. Courseware is a term used to describe the many different types of Computer Assisted Instruction (CAI) resources used in an educational environment. The problem is not in the availability of authoring languages, but in the type of facilities that they can provide. Designers of authoring languages face two major problems; first, to meet the needs of continuing change and update of workstations, and second, courseware transportability between CAL systems does not exist. A software engineering approach to address these problems successfully is described in this paper.

CAL software designers place particular emphasis on the importance of using modular design techniques as the key to writing easy to maintain and portable software. The development of programs and library modules using the Modula-2 programming language are also discussed and examples presented. The successful results of the portability tests that were carried out on BIIVA show that Modula-2 can be used to develop portable software for this application. BIIVA's evaluation results demonstrate that it has good potential for CAL applications.

MÉTHODES DE PRODUCTION DE LOGICIELS POUR APPLICATIONS SUR VIDÉODISQUE INTERACTIF

Cette communication article décrit une méthode de production de logiciels pour applications sur vidéodisque interactif appelée BIIVA (BASIC Interpretor for Interactive Videodisc Applications). On dispose de plusieurs langages de médiatisation pour l'élaboration des didacticiels. Le terme de didacticiel est un terme générique désignant les divers types de logiciels d'enseignement assisté par ordinateur (EAO) utilisés en milieu pédagogique. Le problème ne réside pas au niveau de la disponibilité des langages de médiatisation, mais au niveau des fonctions qu'ils peuvent fournir. Les concepteurs de langages de médiatisation doivent faire face à deux problèmes principaux : d'abord, l'adaptation aux modifications successives ininterrompues des postes de travail et, ensuite, l'absence d'interchangeabilité des didacticiels entre les divers systèmes EAO. Cette communication décrit une méthode de production de logiciels qui s'attaque à ces problèmes.

Les concepteurs de logiciels EAO soulignent l'importance des méthodes modulaires dans l'élaboration de logiciels transférables faciles à tenir à jour. L'élaboration de programmes et de modules de bibliothèque au moyen du langage de programmation Modula-2 est également discutée avec des exemples. Le succès des tests de transférabilité effectués sur le système BIIVA démontre que le langage de programmation Modula-2 peut être utilisé pour mettre au point des logiciels transférables dans ce genre d'application. Les résultats d'évaluation du système BIIVA démontrent que celui-ci est prometteur pour les applications en EAO.

SOFTWARE ENGINEERING TECHNIQUES FOR INTERACTIVE VIDEODISC APPLICATIONS

Dr D. Bell
Dean of Faculty of Engineering
A.K. Anwar
Senior Research Fellow, School of Electronic Engineering

West Glamorgan Institute of Higher Education
Mount Pleasant, Swansea, Wales, SA1 6ED, UK.

Dr P.G. Barker
Reader in Applied Computer & Information Technology
School of Information Engineering
Teesside Polytechnic
Middlesbrough, Cleveland TS1 3BA, UK.

INTRODUCTION

There are three main ways of presenting information: text, sound and pictures. All existing conventional media have limitations in the type of information they can display and the way they can present it. The use of print-based training materials, for example, requires intricate diagrams with laboriously produced text. Computers are used widely in training and to present information, but they are limited only to text and grahpics.

In distinct contrast come the video-based training materials which are concerned with the use of moving pictorial and audio information. There are two basic storage media used for storing video information, these are video tape and video disc.

Howevere, the requirement is for a medium that allows the user to program computer text and graphics together with video pictures and audio, all in a sequence which complements each segment of the subject being taught.

Films and videos, on their own, cannot meet such a requirement because of their transient nature. Time is needed to identify, categorise and mentally organise the presented information.

Interactive Video technology IV is emerging as an effective multi-purpose medium for Computer Assisted Learning CAL applications. IV merges the two powerful technologies of microcomputers and video systems. It utilizes these two technologies in order to present information, to question and to challenge the learner. IV, relies heavily on the successful integration of hardware, software and courseware.

The widening gap between hardware and software development, makes software design for CAL applications a critical area of research. It is a serious challenge that has to be met with the correct software tools. In this paper, we shall be primarily concerned with software engineering techniques that are based upon the use of Modula-2 programming language. The developed software is referred to in this paper as BIIVA, BASIC INTERPRETER FOR INTERACTIVE VIDEODISC APPLICATIONS.

SOFTWARE ENGINEERING and MODULA-2

For the past fifteen years the cost of developing software systems has been steadily increasing while the cost of hardware systems has been decreasing. It was noted then, that programmers were spending an inordinate amount of their effort on testing, debugging and maintaining software. The discipline of software engineering has been developed in response to this situation. It has become apparent that programming languages and software tools must be created to enhance software reliability, streamline the process of software development and promote more efficient software maintenance.

Software engineering was defined as the application of the engineering process to the development of computer software. The key difference from the more traditional approaches is that software engineering stresses integration among the various planning, management and development processes that result in the creation of information systems [1].

However, a successful application of the software engineering approach relies heavily on the development programming language. This language must have the facilities to implement this approach.

CAL software designers have encountered serious difficulties when using programming languages such as FORTRAN. This language uses the 'GOTO' statement which leads to hard-to- follow constructs. FORTRAN was used to design CAL software system by the Assisted Teaching unit, Queen Mary College, London, UK [2]. FORTRAN was also used to develop libraries of subroutines for CAL applications by Ulster Polytechnic, North Irland, UK [3].

The technique of structured programming, on the other hand, can be a useful tool for writing easy-to-follow programs. However, this technique does not help to divide the programming task into manageable pieces. PASCAL programming language was used by Delf University of Technology- The Netherlands, to design a portable software system to present courseware on microcomputers [4].

The importance of using modular design, was stressed by the research projects mentioned above. Emphasis was also placed on this particular design approach as the key to writing easy to maintain and portable software. The need for a suitable programming language that can implement the modular design approach efficiently, was clear.

THE STRUCTURE OF MODULA-2

One of the most important features of Modula-2 programming language is the module structure. A module can be described as a collection of declarations that work together to perform a given task. Each module is an autonomous code unit which can be separately specified, developed, compiled and tested [5]. There are two different types of modules in Modula-2:

(1) Program module: Which is a complete program unit that may consists of one or more modules and use facilities from library modules.
(2) Library module: It is used to export facilities to other modules. These other modules may be part of a particular software package or may be modules in other separate software systems. Each library module has two parts: definition module and implementation module.

SEPARATE COMPILATION

This has traditionally been a great source of problems, especially when parts of a large software system are developed by different people. Modula-2 solves this problem by maintaining a system-wide module-version control scheme [6].

The separate compilation technique has great implications on multiperson project management for the following reasons [7]:

(1) The system's definition modules can be distributed in compiled form to the programming team members, who can then write and modify implementation modules without forcing everybody else on the project to recompile when a change is made.
(2) Individual team members cannot alter a module interface, as the compiler enforces conformance between the procedure heading declared in the definition module and the corresponding headings in the team member's implementation module.
(3) If team members are granted access to the source text of definition modules the version control system prevents the execution of mismatched modules.

COURSEWARE TRANSPORTABILITY

Courseware is a term used to describe the many different types of Computer Assisted Instructional CAI resources used in an educational environment [8]. Courseware transportability has serious implications not only on the effectiveness and quality of the developed courseware, but on the future of CAL as a whole. The availability of transportable coursware can encourage the inter- institutional software exchange, and makes it more feasible. It can ensure a more efficient and cost-effective strategy for CAL software development. The ease of courseware development will depend upon two major factors:

(1) The types of resource that the CAL workstation utilises.
(2) The suitability of the available software development tools [9].
Courseware preparation can be achieved by using either conventional programming languages or authoring languages. Designers of authoring languages face major problems:

(1) CAL workstations have to be changed and updated to meet new requirements that can be caused by technological, economical and organizational developments.
(2) Courseware transportability dose not exist between CAL systems [4].
The structures of authoring and conventional programming languages are so different that it is impractical to construct conversion programs. Typical facilities that CAL programs may exploit are usually the facilities that are sacrificed in the interest of courseware transportability. This is because features may prove to be dependent upon a particular computer or operating system. There is no standard high level complex machine independent authoring language exists.

SOFTWARE PORTABILITY

Software for CAL applications to meet the requirements of most disciplines and suitable for portability between different computers is not yet available.

Two main reasons were given for this problem:

(1) The continuous use of conventional software design approaches.
(2) The use of unsuitable development programming languages.

Ideally speaking, a program should be portable over a range of different computers without alterations so that it compiles and runs satisfactorily. In reality when transporting a program between computers, some changes are usually necessary before a satisfactory run is obtained. However, a program can still be called portable, if the modifications that are needed to achieve a satisfactory run on each computer are easy and the resulting program is recognisably similar to the original [10].

Unlike other Interpreters BIIVA has been designed to be portable. It was completely written in Modula-2 programming language. There was no need to write any part of the program in the assembly language of the specific microcomputer. This is essential for designing software for maximum portability. This could not have been easily achieved without Modula-2's library concept. A set of standard modules provided BIIVA with an operating environment that is independent of the operating system.

A full report of the portability test on BIIVA was provided by Real Time Associates. It is the software specialist company that carried out the test and verified its results.

The significance of this test, is that it covered different range of Modula-2 compilers, computers and operating systems. The computers used in the test were, AMISTRAD PC, APPLE MACINTOSH and DEC PDP- 11.

The test shows clearly that Modula-2 programming language has the potential for developing portable software that can be moved, with little or no change, across different computers and operating systems. The successful results of the test were a step in the right direction in designing standard stand alone machine independent code.

Testing the transportability of courseware developed by BIIVA was not possible. It was prevented by the lack of resources that would be needed to setup interactive videodisc stations for different computers and different environments. Nevertheless, testing the portability of BIIVA successfully was the most important step, since the courseware in question is developed using BIIVA itself.

THE STRUCTURE OF BIIVA

BIIVA is designed to be flexible and easy to maintain and update. It is divided into a main program module and several library modules. These modules are interconnected to each other via the Modula-2 system's environment.

These modules include procedures to support editing facilities and arithmetic operations. BIIVA supports two types of commands:
(1) The videodisc player control commands, such as PLAY and STOP. These commands were designed as library modules for maximum flexibility.
(2) Other commands such as PRINT, GOTO and IF. They are used to perform arithmetical functions and to control the flow of program.

BIIVA supports the use of mouse as the user interface.

BIIVA executes its commands in a similar way to the BASIC programming language.
(1) direct mode: this involves typing the desired command and pressing <CR>.

(2) Indirect mode: this involves writing a program. In this case the command RUN must be used to execute the developed program.

BIIVA IMPLEMENTATION

The workstation that was use to develop and test BIIVA will be used here to implemented it. The minimum hardware requirements are as follows:
(1) IBM-PC, XT, AT or compatibles with two RS232 serial ports to drive both, the videodisc player and the mouse. BIIVA requires 30KB of RAM.

(2) Mouse: Microsoft, Logitech or compatibles. The mouse driver must be installed before using BIIVA.
(3) Overlay Interface:
BIIVA can be used with any general purpose overlay interface, provided that it does not need software to control the videodisc player. The overlay interface that was used with this workstation is the Video Integrator.
(4) Videodisc Player: BIIVA can only drive the Philips VP 835 module.

BIIVA supports both, teacher's environment and student environment. It does not monitor student responses.

EVALUATION

In order to assess the usability of BIIVA, an evaluation was carried out on its capabilities. A combination of guided study and questionnaires was employed.

This evaluation was carried out at the School of Electronic Engineering and the results were verified

by the research committee of the Faculty of Engineering, West Glamorgan Institute of Higher Education, Swansea. Several members of staff in the Institute and form outside have used the interactive videodisc workstation and BIIVA. Each user has carried out the following tasks:

(1) Used BIIVA to go through a videodisc.
(2) Prepared courseware materials.
(3) Test the effectiveness of the prepared courseware on two students.
(4) Completed the provided questionnaires.

The evaluation results proved that BIIVA is easy to use and it took less than one hour to become familiar with it. Also it has a good potential to develop simple courseware materials for CAL applications. This results also showed that BIIVA and the IV workstation would be used by lecturers as a teaching method if they are available.

FUTURE WORK

The future work can be seen in three major areas:

(1) BIIVA can be converted from an interpreter to a compiler. This will improve its efficiency, reliability and speed.
(2) BIIVA can be designed as an embedded software pakage. It can be down loaded to different target microprocessor which can be mounted on an external naked board, Intel or Motorola. Then it can be addressed from the host computer using different operating systems.
(3) BIIVA can be improved in the following areas:
(1) To handle students response and to support database systems.
(2) Restructuring BIIVA so that all commands become library modules. This can increase its flexibility, ease of update and maintenance.
(3) Adding PopUp menus to the mouse module, so that the videodisc control commands can be selected from these menus.

CONCLUSIONS:

This paper describes the use of software engineering techniques for interactive Videodisk applications. The developed software is referred to in this paper as BIIVA, BASIC INTERPRETER FOR INTERACTIVE VIDEODISC APPLICATIONS.

There are many authoring languages available for courseware development. But the problem is not about, how many are available as much as the type of facilities that they can provide. Other important factors include, user friendliness, courseware transportability and the ease of interfacing to other operating systems.

Designers of authoring languages face two major problems:
(1) to meet the needs of continuing changes and update of workstations.
(2) courseware transportability between CAL systems does not exist.

It was fully realised that the claimed loss in effectiveness was not inherited in the concept of CAL, but in the inadequacy of the software development tools used. Software engineering approach to CAL software design using Modula-2 programming language, seems to address these problems successfully. Emphasis was placed on the use of modular design approach.

Facilities offered by BIIVA are less powerful than those being offered by known authoring systems, and yet there seem good reasons to believe that it should be suitable for a wide range of interactive videodisc applications.

The effort and time needed to become fluent in its use, is likely to be less that that needed for other authoring languages. It may therefore also form a useful transitional step for those who will ultimately need to use authoring systems, since it introduces many concepts of this subject.

The portability test, which was carried out on BIIVA, showed clearly that Modula-2 programming language can be used for developing software that can be transported, with little or no change, across different computers, operating systems and environments. The successful results of the portability test, is a step in the right direction in designing stand-alone machine independent code.

REEFERENCES

(1) CASE, A.F. , International Systems Development, principles of computer-aided software engineering, ISBN 0-13-464520-0, Prentice-Hall, 1986.
(2) BLANDFORD, C. , CAL Software Design For Transfersbility, Computers & Education, Vol. 6, pp. 165-174, 1982.
(3) WILSON, R.N. & McCRUM E. , Use of Modular Design in The Production of CAL Software: A Case Study, Computers & Education, Vol. 8, pp. 229-237, 1984.
(4) MAST, C., A Portable Program to Present Courseware on Microcomputers, Computers & Education, Vol. 6, pp. 39-44, 1982.
(5) WIRTH, N. , Programming in Modula-2, ISNB 0-3-540-15078-1, Springer-Verlag, 1982.
(6) FORD, G.A. & Wiener R.S. , Modula-2; A Software Development Approach, ISBN 0-471-87834-0, Joh Wiley & Sons Inc. , 1985.
(7) McCORMACK, J. , & GLEAVENS, R., Modula-2 a Worthy Successor to Pascal, BYTE, Vol. 8, No.4, pp. 385-395, April 1983.
(8) BARKER, P.G., & SINGH, R., Authoring Languages for Computer Based Learning, British Journal of Educational Technology, Vol. 13, No.3, pp. 167-196, October 1982.
(9) BARKER, P.G., & YEATES, H., Introducing Computer Assisted Learning, ISBN 0-13-477308-x, Prentice-Hall International , 1985.
(10) BROWN, J., Software Portability, Cambridge Computers Science Texts, 1977.

ACQUISITION DE NOTIONS ET CHANGEMENT D'ATTITUDE: UNE STRATÉGIE D'EXPLOITATION DU VIDÉODISQUE INTERACTIF

M. Decorte, J. Morissette
Centre d'enseignement et de recherche en informatique
Clément-Lockquell Inc.
Sainte-Foy, PQ

En 1988, le Centre d'enseignement et de recherche en informatique Clément-Lockquell Inc. a réalisé une activité de formation vidéo-interactive de niveau 3 portant sur le processus d'entrevue-conseil. Cette activité de formation a été expérimentée auprès du personnel d'une vingtaine de Caisses populaires.

Nous énumérerons les étapes de développement du projet. Nous allons décrire comment nous avons exploité les possibilités de la vidéo-interactive pour atteindre les objectifs de la formation (aspects "design" et ressources matérielles).

Les outils informatiques développés par le Centre pour produire l'activité de formation seront présentés, ainsi que la méthode de développement de l'activité de formation.

Une période de questions est prévue à la fin de la démonstration.

ACQUISITION OF CONCEPTS AND CHANGE OF ATTITUDE: A STRATEGY FOR USING THE INTERACTIVE VIDEODISC

In 1988, the Centre d'enseignement et de recherche en informatique Clément-Lockquell Inc. worked on a project on level 3 interactive video training dealing with the counselling interview process. This training activity was tested with the staffs of twenty or so Caisses populaires.

We will list the development stages of the project and describe how we made use of the possibilities of interactive video to achieve the training objectives (design and material resource aspects).

The computer tools developed by the Centre to produce the training activity will be presented, as well as the method for developing the training activity.

A question period is planned at the end of the demonstration.

Acquisition de notions et changement d'attitude: une stratégie d'exploitation du vidéodisque interactif

Michelle Decorte et Jules Morissette

Centre d'enseignement et de recherche en informatique Clément-Lockquell inc.

1. Résumé

De décembre 1987 à janvier 1989, le Centre d'enseignement et de recherche en informatique Clément-Lockquell inc. du Cégep de Sainte-Foy développait, en collaboration avec la Confédération des caisses populaires et d'économie Desjardins du Québec, une activité de formation vidéo-interactive portant sur le processus d'entrevue conseil.

Il s'agit d'une activité de niveau 3 développée sur Infowindow d'IBM utilisant l'écran tactile comme périphérique d'entrée.

Ce projet, subventionné par Emploi et Immigration Canada dans le cadre du programme national d'aide à l'innovation, vise à démontrer l'efficacité de la vidéo-interactive pour l'apprentissage de notions abstraites.

Nous vous présentons ici la description des objectifs de la formation, la structure de l'activité, les stratégies pédagogiques utilisées et les moyens mis en oeuvre pour évaluer l'efficacité de cette activité de formation.

2. Objectifs de la formation

La Confédération des caisses populaires et d'économie Desjardins du Québec veut former ses agents conseils (actuels et futurs) à un nouveau processus d'entrevue. Dans le processus actuel, l'agent conseil se limite en général à répondre à la demande telle que formulée par le client.

Le nouveau processus, l'entrevue conseil, vise à élargir le cadre de l'entrevue afin d'analyser les besoins financiers du client de façon globale.

L'activité de formation qui fait l'objet du présent projet ne constitue que la première étape du programme de formation.

De façon plus spécifique, au terme de cette activité l'agent conseil devrait être en mesure d'identifier les étapes et les comportements du processus et de procéder à l'auto-évaluation de ses comportements.

3. Structure générale de l'activité

L'activité de formation comporte les huit modules suivants:

1. FAMILIARISATION
2. INTRODUCTION
3. ACCUEIL
4. EXPLORATION
5. EXAMEN
6. CONCLUSION
7. VISIONNEMENT
8. LABORATOIRE

Ces huit titres figurent au menu principal qui permet à l'agent conseil d'accéder au module de son choix.

Le premier module vise à familiariser le participant avec l'outil de formation.

L'introduction présente la problématique, le processus d'entrevue conseil et les objectifs de formation.

Les options 3,4, 5 et 6 correspondent aux quatre étapes du processus d'entrevue conseil. Chacun de ces modules présente les notions et les exercices relatifs à l'étape concernée.

L'option visionnement permet de voir le déroulement de deux exemples d'entrevues conseils, du début à la fin.

Le laboratoire constitue le test final de l'activité.

4. Stratégies d'exploitation du vidéodisque interactif

Les leçons qui constituent les différents modules de l'activité de formation exploitent au maximum les possibilités de la vidéo-interactive. Les concepts théoriques y sont présentés par un enchaînement dynamique de séquences tutorielles et de simulations vidéo.

Séquences tutorielles:

Les notions théoriques sont présentées par une narratrice qui joue un rôle central dans l'activité. Elle présente les notions par des tutoriels vidéo où elle apparaît installée dans un restaurant, ou encore en voix off lorsque le résumé des notions est affiché à l'écran.

Le contexte du restaurant dans lequel la narratrice est placée permet, entre autres, de rendre plus dynamique l'exposé des notions.

Simulations vidéo:

Les simulations enrichissent la théorie par les mises en situation qu'elles présentent et par la possibilité qu'elles offrent de rendre une ambiance et des émotions. On peut mettre en valeur la relation de confiance qui doit s'établir entre le client et l'agent conseil. On peut suggérer des façons (comportements verbaux et non-verbaux) d'aborder certains sujets délicats.

Trois types de simulations sont utilisées:

A) Simulations d'entrevues conseils:

Une simulation d'entrevue entre un agent conseil et un sociétaire illustre les notions presentées par la narratrice. Cette simulation est découpée en 16 séquences qui sont intercalées dans les leçons au moment opportun avec surimpression éventuelle d'information à l'écran. Deux autres simulations d'entrevue conseil sont utilisées pour les exercices vidéo-interactifs.

L'option "visionnement" du menu principal offre la possibilité de voir complètement et sans interruption chacune de ces entrevues. De plus, lors du visionnement le comportement en cours est identifié au bas de l'écran.

Le réalisme des entrevues présentées amène l'agent conseil à se reconnaître dans ces mises en situations. Cela favorise le rappel des notions apprises lorsqu'il se retrouve dans les situations réelles de son travail. Il dispose alors d'un point de comparaison lui permettant de s'auto-évaluer.

B) Illustration de comportements dans le restaurant:

Les notions théoriques associées à certains comportements de l'entrevue conseil sont illustrées dans le restaurant par le jeu des serveurs et des clients, ce qui apporte un élément dynamique supplémentaire.

C) Présentation de la problématique:

La problématique qui a mené à l'élaboration de cette activité de formation est présentée dans l'introduction par une séquence vidéo dont le scénario à été particulièrement étudié pour faire ressortir les facteurs qui justifient la formation des agents conseils, pour motiver l'agent conseil à suivre l'activité de formation et aussi pour présenter les différents intervenants (narratrice, agents conseils, sociétaires, serveurs du restaurant ...).

Exercices d'apprentissage:

A la fin de chaque module de formation, l'agent conseil est invité à réaliser un certain nombre d'exercices (douze au total).

A) Exercices vidéo-interactifs:

Ce type d'exercices vise essentiellement à vérifier la capacité de l'agent conseil à identifier les comportements de l'étape de l'entrevue conseil sur laquelle porte le module de formation.

Il s'agit par exemple:

- de repérer la présence ou l'absence d'un comportement dans une simulation vidéo;
- de reconnaître la bonne interprétation d'un comportement parmi plusieurs séquences vidéo;
- d'associer un ou des commentaires à une séquence vidéo;

Les simulations vidéo utilisées dans ces exercices sont des extraits d'une entrevue conseil différente de celle utilisée pour illustrer les notions théoriques dans les leçons. Ces exercices exploitent tout le potentiel de la vidéo interactive.

B) Exercices interactifs de type "vrai ou faux" et "associations":

Ces exercices permettent à l'agent conseil d'évaluer ses connaissances. De plus, les indices affichés à l'écran pour l'aider à corriger ses erreurs renforcent les notions théoriques.

L'ordre des questions et des choix de réponses est aléatoire chaque fois qu'on accède à un exercice.

Le soin apporté à la qualité esthétique des pages écrans, la diversité des exercices et la rétroaction immédiate contribuent au maintien de la motivation de l'agent conseil.

C) Le laboratoire:

Ce module est accessible par le menu principal.

L'élément principal du module est un test lors duquel on invite l'agent conseil à interrompre l'entrevue afin d'identifier le comportement en cours au moment de l'interruption.

Ce test est d'autant plus intéressant que l'ordre dans lequel l'agent conseil adopte les comportements de l'entrevue conseil est différent de l'ordre séquentiel suggéré jusqu'a ce moment par la théorie. Cette différence n'affecte pourtant pas la qualité de l'entrevue, il est au contraire justifié de procéder ainsi dans le contexte de la demande de ce client.

L'activité de formation se termine donc par une invitation à appliquer avec réalisme les notions théoriques présentées.

5. Gestion informatisée de l'apprentissage

Lors de son premier passage dans l'activité, l'agent conseil est obligé de parcourir les modules et les leçons dans un ordre pré-établi. Par la suite, il peut accéder au module où à la leçon de son choix par le biais d'une organisation hiérarchique de menus.

Une barre de commandes composée de 7 touches tactiles située au bas de l'écran permet à l'agent conseil de cheminer à son rythme dans une leçon (recommencer au début, revoir la page ou la séquence précédente, passer à la page ou à la séquence suivante, demander de l'aide, quitter la leçon ou le module).

Le dossier de l'agent conseil est mis à jour au fur et à mesure de son cheminement.

6. Expérimentation et évaluation de l'activité

Afin d'évaluer l'efficacité de cette activité de formation, trois postes vidéo-interactifs seront installés pendant une période de deux semaines dans quinze à vingt Caisses populaires Desjardins et l'activité sera expérimentée dans un contexte d'autoformation auprès d'une quarantaine d'agents conseil.

Deux personnes-ressources du Centre Canadien de Recherche sur l'Informatisation du Travail se sont jointes au projet pour la préparation de l'expérimentation. Ils ont proposé une stratégie d'expérimentation et ont développé les questionnaires à compléter par les participants avant et après la formation. Huit à onze semaines après la formation, on rencontrera les participants afin d'évaluer la qualité du transfert des acquis dans la pratique de leurs entrevues. Le contexte d'implantation de l'activité dans le milieu de travail sera considéré comme un paramètre important pouvant influencer l'atteinte des objectifs de la formation.

On mesurera aussi les qualités du médium quant à son efficacité pour susciter et maintenir la motivation de l'agent conseil.

A la suite de cette expérimentation, le Centre d'enseignement et de recherche en informatique Clément-Lockquell inc. diffusera trois rapports portant respectivement sur la pertinence de l'utilisation d'une activité vidéo-interactive, sur l'estimation du rapport coûts/efficacité d'une telle activité et sur la méthodologie de développement suggérée.

IMPLEMENTING INTERACTIVE VIDEODISC INSTRUCTION

G. Richards
IBM/BCIT Interactive Videodisc Development Project
British Columbia Institute of Technology
Burnaby, BC

Since March of 1987, the British Columbia Institute of Technology has embarked on a project to produce interactive videodiscs and introduce them into health sciences classrooms and study centres on the BCIT campus. Key to the success of the project has been the involvement of faculty in the design and development of the courseware supported by a small cadre of experienced CAL and interactive video producers. A variety of approaches have been tried - including adopting outside materials, producing comprehensive tutorial packages, data-base projects, and involving BCIT in a sharedisc production. In terms of acceptance by faculty and students, the results have been equally varied and the lessons learned in organization and scale of production are now being applied to on-going productions. Segments of "The Anatomy and Physiology of the Heart" will be used to illustrate the presentation.

MISE EN OEUVRE D'UN ENSEIGNEMENT AVEC VIDÉODISQUES INTERACTIFS

Depuis mars 1987, la British Columbia Institute of Technology s'est engagée dans un projet de production de vidéodisques interactifs et d'intégration de ces derniers dans les classes de science de la santé et les centres d'étude de son campus. La clé de voûte de ce projet a été l'implication du corps professoral dans la conception et le développement du didacticiel, appuyé par un petit groupe de spécialistes de l'EAO et de producteurs de vidéo interactive. Plusieurs types d'approche ont été testés - y compris le recours à du matériel extérieur, la production de tutoriels complets, des projets de base de données et la participation du BCIT à une production de disques conjoints. Les réactions des professeurs et des étudiants ont été tout aussi variées; l'expérience acquise au sujet de l'organisation et de l'échelle de production sert maintenant dans les productions en cours. On se sert de séquences de la production "The Anatomy and Physiology of the Heart" pour illustrer le présent exposé.

IMPLEMENTING INTERACTIVE VIDEODISC INSTRUCTION

Griff Richards

Project Leader,
IBM/BCIT Interactive Videodisc Development Project
British Columbia Institute of Technology, Burnaby, B.C.

Introduction

Training by interactive video has rapidly caught on in industrial and military settings where the emphasis is on efficiency and speed of training and standardization of skills. Interactive video has made less progress in educational environments where there is a heavy commitment to traditional classroom instruction and where reducing total training time may not generate incentives to warrant investment in high-technology training systems.

In March, 1987, The British Columbia Institute of Technology (BCIT) and IBM Canada Ltd. began work on a joint project to explore the potential of interactive videodisc technology for application to the process of teaching and learning. The project is the outgrowth of IBM's use of InfoWindow videodisc systems at Expo '86 and receives funding from the British Columbia Fund for Excellence in Education.

To be successful, implementation of interactive videodisc instruction in an academic environment must overcome a number of hurdles. The object of this paper is to discuss the implementation approach taken by the IBM/BCIT IVD Project.

Finding the useful software

In many respects, with hardware and funding in place, the IBM/BCIT project had already overcome the two major hurdles faced by most academic institutions. However, hardware without software is a long way from IVD implementation.

While the list of published videodiscs has grown tremendously in the last 5 years, the list of discs that might be appropriate for use in BCIT's School of Health Sciences was quite small. Of the few that were selected for preview, only two were purchased. Of these, neither has been actually used in teaching. A third disc was identified as having potential use for nursing instruction however, the faculty felt that the $US 1000 could be better spent on other learning aids.

This does not imply that faculty are resistant to using videodiscs. On the contrary, in the Sheet Metal Program at BCIT videodisc-based materials have been used for almost eight years to enhance classroom instruction. Thus, success lies in the usefulness of the messages rather than the medium itself. The decisions to buy or not buy videodiscs were made on the criteria of relative usefulness when compared with other options. It became clear, that for any innovation to be successfully implemented, it would have to be useful.

Striving for Usefulness

Finding useful courseware was not a problem unique to BCIT. Indeed, recognition of the paucity of materials led to the establishment of the IVD Project goals requiring the production of two videodiscs for healthcare instruction. These discs would not only test IVD production methodology, but would make a contribution to the pool of videodisc materials available for first-year health science curriculum.

The development of interactive video can be expensive. It is an inter-disciplinary undertaking requiring expertise in the areas of instructional design, audiovisual production, software programming and computer hardware in addition to expert knowledge of the subject matter to be taught. All of the above expense comes to naught if the finished program is only usable by one user and in the end is never distributed nor used.

Usefulness implies not only having the right information, but organizing it in a fashion consistent with the needs of a program's potential users, both learners and instructors. Usefulness is achieved when potential users track down the project, buy it, use it and tell others how good it is. Thus, the realization of a project begins not with the producer saying "this is what we can do", but with the users saying "this is what we need".

Implementation implies buy-in and on-going support

From the start, the architects of the IBM/BCIT IVD Project were concerned that the focus be on the total implementation rather than merely aspects of technical production. It was felt that a stronger implementation - one lasting beyond the tenure of a three-year project - would be achieved by establishing support from existing groups on campus.
For this reason, instructors were given term-sized blocks of release time to actively participate in every stage of the development of interactive videodiscs. This meant that discs would be produced to address problem areas identified by the instructors themselves thus ensuring some level of acceptance in the faculty.

Secondly, where possible, videodiscs should have some "stand-alone" merit so they could be used as a source of visuals in the classroom. This flexibility meant instructors could use the disc without having to buy into the constraints of a computer-assisted learning system. This greatly simplified the logistics for videodiscs in the classroom and did not preclude the use of a computer-aided learning system in the learning resource area. With fewer "barriers to use", it is hoped that more instructors could benefit from the videodisc resource and that the positive attitude towards videodisc might translate into interest in further projects.

Thirdly, the IVD project would piggy-back on other traditional sources of media which existed within the institute and thus the project was centred within the Learning Resources Unit with access to technical expertise in audio-visual production as required. Some project monies were used to upgrade BCIT's production facilities in exchange for production services.

Finally, if in-house resources were not available, the contracting-out of services was done in such a way that, wherever possible,

BCIT staff would have the opportunity to learn though working with local experts. This provided an opportunity for staff development and developed resources for in-house production of videodiscs.

The size and prestige of the project brought a good deal of support at different levels in the institute. A number of project advisory committees provided information flow during the early stages of the project to administrators, faculty and co-operating hospitals and other concerned agencies. During early phases of development a significant number of tours and demonstrations were given to visitors to BCIT. Indeed, as the project moved into the production phase, the tours were curtailed as they interfered with the project timelines. In retrospect, project timelines should have made a larger allowance for demonstration time since there is a certain amount of truth to the maxim "Out of sight - out of mind.

The implementation is more than software development

"So what do I do with this videodisc? How do I show it to my classes?" was a major concern of a faculty member of our team. The medical imaging clips that had been requested had been pressed on disc for easy control and stop action etcetera, but he wanted to know how to incorporate these materials into the instructional process.

Four potential ways to use the videodisc were seen and plans were made for the hardware support of each:

1. The classroom is fitted with a videodisc player and either a large monitor or a video projector. If need be, these units can be placed on a cart and wheeled from site to site. Using a remote control unit or a mouse-type pointing device, the instructor can access the information on the videodisc for display during class.

2. The learning lab contains 10 interactive video learning stations and it can be booked by an instructor to bring a group of up to 20 students in during class. This can be particularly effective in early class sessions, but requires a copy of the videodisc for each learning station.

3. The computer centre is a central location for student use of a number of microcomputers. A carrell for videodisc instruction can be accessed by individual students for use outside of normal class hours. This scenario is especially important for remedial instruction or for self-paced instruction.

4. The off-campus site is proposed for access in hospitals or satellite campuses. This could simply be a number of delivery systems which are rotated form site to site installed on a permanent basis in the hospitals. Three Levels of Project Scope Proposed

Project Scope

As work progressed on the first videodisc - a large encyclopedic work entitled "The Anatomy and Physiology of the Heart" - a smaller quick project was completed putting together a collection of medical radiographs (X-rays) and transferring this database onto videodisc. Eventually this material was contributed to the University of Utah's sharedisc, "The Third Slice of Life ".
Through participation in these projects, it became readily apparent that the project methodology required a new set of rules regarding levels of support and communications if it were to encompass videodisc collections and co-operative projects.

Interactive videodisc development within an institution will only progress if the level of commitment by the institute is matched by the enthusiasm of the instructors and students. Nothing dampens enthusiasm faster than involvement in a project beyond the capabilities, interest and resources of the project team.

For this reason, three levels of production are proposed:

A Mega project involves the production of a complete interactive videodisc program. All phases of design, video production, computer programming, evaluation, revision and marketing are conducted. The result is an educational package - either a simulation or tutorial and the cost is between $150K to $250K. A Mega project would probably take 9 to 24 months to complete.

A Mini project involves the production of a videodisc without the accompanying computer tutorial. Collections of images are assembled and catalogued or existing materials such as "Welcome to BCIT" are transferred to disc and a database assembled. Some mini projects might have a small user interface program but in all, the total budget would be from $50K to $150K and the project take from 3 to 9 months. Pilot projects for Mega projects would be classed as Mini projects.

Micro projects involve the production of a testbank or tutorial around videodiscs from earlier projects or purchased. Because there is no video production, costs and timelines are considerably shorter - from $10K to $50K over a 1 to 6 month period.

Examples:

The Anatomy and Physiology of the Heart would be classed a Mega project. A project with about eight to ten hours of interactive instructional material probably cost about $200,000 and it will have taken about 20 months from concept to delivery. This level of involvement and activity cannot continue without a significant flow of grant money so the product should be addressed at a market of sufficient size or importance so that costs can be recovered. But how many institutions can or will pay $1000 to $2000 for a copy of the package?

The Medical Radiography videodisc is a Mini project. Total cost was about $10K with much of the work being done by students on work grants or summer subsidies. Pressing and transfer costs ran about $3,000. The total project was completed in less than a year and eventually the final project was rolled into a sharedisc for lower pressing costs and other considerations. The disc itself does not have an IVD tutorial but is a database which can be used in a number of ways.

A Micro project has not yet been done within the IVD project, but if a faculty member designed a testbank around the materials on the medical radiography videodisc, it could be classed as a Mini project. The cost would vary depending upon the amount of support required and whether faculty release time was funded as opposed to a student work-study program. The advantages of such a project are the short turn-around time, the low cost and the immediate applicability of the materials to a course at BCIT.

Institutes wishing to implement interactive videodisc instruction might be well advised to become involved in a number of small, easily managed micro projects before plunging into the complexities of mini and macro scale videodisc productions.

Student participation in project development

BCIT students have been hired during summer periods with good success after relatively short training periods. One student was recently hired by a local production house to work on IVD projects which, while not suggesting a large market, does suggest the recognition of marketable skills in the IVD area.

The academic week at BCIT provides little time for students to be involved in part-time IVD production during the academic term, however if they are employed in tasks directly related to their studies, the rewards can be tremendous. As one student remarked about the radiographic project "It's like being paid to do your homework."

Given proper supervision by faculty and developers, the enthusiasm and creativity of students can be most productive and rewarding.

Continuing support for IVD activities

I once asked a faculty member of a course development team at the Open University if subsequent projects would run smoother than pilot projects. He replied "Who would ever be so daft as to go through this process twice?" Therein lies the constraint of relying solely on interested faculty to develop complex learning materials.

While the subject-matter expertise of instructors and their participation in the development process is essential for a successful project, the primary function of instructors is to instruct. Even given release time to work on a project, eventually the instructors return to their classrooms taking with them new knowledge and new, useful videodiscs.

As long as the process of developing interactive videodiscs remains complex and time-consuming, the burden for on-going development will rest with the cadre of professional developers who:

1. take the necessary steps to include faculty and other stake-holders whenever possible,

2. fit the size and scope of the project to the time and resources available for production,

3. encourage new players to get involved through incremental increases in project complexity,

4. ensure that the interactive videodisc materials are useful and

5. plan the implementation as well as the production.

Acknowledgement

Much of the IVD implementation planning was set out before the author's participation in this project by Marg Penney (currently at Fraser Valley College) and Ron Graham (IBM Canada Ltd.).

Reference

Penney M., Richards G. and Fenrich P. Interactive Videodisc Project Methodology Report, BCIT, January, 1988.

DESIGNING A COMPUTER VIDEODISC SYSTEM TO TRAIN FOREST FIRE CONTROL PERSONNEL

L.C. Emmott
Educational Technology Centre
ACCESS NETWORK
Calgary, AB

ACCESS NETWORK, The Alberta Educational Communications Corporation, and Alberta Energy and Natural Resources, through the Forest Technology School in Hinton, Alberta, are producing a computer videodisc system to train forest fire control personnel in initial wildfire assessment and attack. The system combines real-life forest fire visuals and dramatized scenarios on videodisc and computer graphic simulation with computer assisted learning courseware to produce an interactive training system. Learners are first guided through a simulated fire to learn the basics of initial assessment and attack. They then are provided with an opportunity to assess and control a simulated wildfire by deploying a variety of suppression resources against the fire. Success is measured by the natural resources saved versus the number of resources used to control the wildfire. This paper will focus on the design process using the expertise of instructional designers, content experts, interactive systems designers, and computer programmers to produce an interactive training system. The author will trace the development of this training system from its inception to the final production. Through lecture and demonstration, the author will describe the process used emphasizing the variety of media used to produce a dynamic and realistic training system.

CONCEPTION D'UN SYSTÈME DE VIDÉODISQUE PILOTÉ PAR ORDINATEUR POUR LA FORMATION DU PERSONNEL DE PROTECTION CONTRE LES INCENDIES DE FORÊTS

ACCESS NETWORK, l'Educational Communications Corporation d'Alberta et la Energy and Natural Resources d'Alberta, par l'entremise de la Forest Technology School de Hinton, Alberta, sont en train de produire un système de vidéodisque piloté par ordinateur qui servira à former le personnel chargé de la protection contre les incendies de forêt pour ce qui est de l'évaluation initiale des incendies et de la lutte contre ces derniers. Le système comporte des images réelles d'incendies de forêt, des scénarios fictifs sur vidéodisque, une simulation graphique par ordinateur de même qu'un didacticiel d'EAO qui constitue un système de formation interactif. Le système présente d'abord un feu simulé et permet d'apprendre les fondements de l'évaluation initiale et de la répression de l'incendie. Il est alors possible d'évaluer et de combattre un incendie simulé en faisant appel à diverses ressources. Le succès obtenue se mesure par l'importance des ressources naturelles préservées par rapport au nombre de ressources utilisées pour combattre l'incendie. Dans le présent document, nous allons nous intéresser au processus de conception dans lequel on a fait appel à l'expérience de concepteurs d'outils éducatifs, d'experts en contenu, de concepteurs de systèmes interactifs et de programmeurs pour produire un système de formation interactif. L'auteur retracera le plan d'élaboration de ce système de son point de départ jusqu'à sa production. Par un exposé et une démonstration, il décrira la démarche qui a été suivie, mettant en évidence les divers moyens utilisés pour produire un système de formation dynamique et réaliste.

DESIGNING A COMPUTER VIDEODISC SYSTEM TO TRAIN FOREST FIRE CONTROL PERSONNEL

Lyle C. Emmott

Educational Technology Centre
ACCESS NETWORK
Calgary, Alberta

Introduction

Every year approximately 300 000 hectares of Alberta forest are lost to wildfire. Minimizing this loss, which is the responsibility of the Forest Protection Branch in the Alberta Forest Service, requires a program of active prevention, immediate detection, and early suppression through initial attack.

Providing one of the best wildfire fighting programs in the country depends, in part, on moving highly trained initial attack crews to the fire scene immediately. Prompt and accurate analysis and assessment of a wildfire are crucial to a successful wildfire suppression program. Thus, experienced and well-trained firefighters who can assess the wildfire quickly are needed. The accuracy of their assessment is an important factor in maintaining a cost-effective suppression program. For that, a training system that provides these professionals with the knowledge and skills to assess and suppress wildfire effectively is essential.

Although the Forest Technology School (FTS) in Hinton, Alberta, provides training for Alberta's forest firefighters, until now it has had no formal wildfire assessment course. Instead, on-the-job training and role play simulations were used for basic instruction. In 1986, the FTS commissioned a study to determine how the school might improve wildfire assessment training. The outcome of that study was a recommendation that the FTS change its instructional delivery system to one that utilized computer based training and interactive videodisc.

The FTS asked ACCESS NETWORK's Educational Technology Centre (ETC) to design a training system to meet the school's immediate and long-term training needs. After preliminary study, the ETC began development in January 1988. This paper describes the project, emphasizing the design process used to develop the computer videodisc system.

Task Analysis

The task analysis is one of the most important steps in the design process. The project type determines the analysis used. Where specific curriculum exists, the task analysis itemizes only those areas of the curriculum that apply to the project. Without a curriculum, a formal task analysis is undertaken to determine the specific tasks, who performs them, the task outcomes, the performance skills and knowledge, the task interrelationships, and the required tools or

references for each task. Because FTS had no curriculum, a formal task analysis was conducted.

During this phase of the project, a wildfire assessment simulation model was developed. It described each task and mapped the assessment process from the time before a fire call to the final mop up. Figure 1 illustrates a portion of this model. Detailed task sheets were prepared from this model to describe each task and subtask, the skills and knowledge required to accomplish each task, as well as ratings of the frequency of performance, the importance of the task, and the performance and learning difficulty. The information gathered during this phase was used to analyze the content in the next phase.

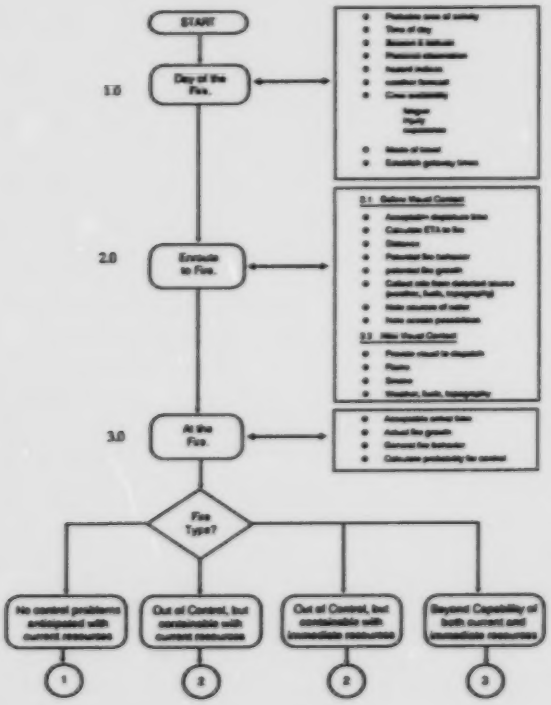


Figure 1: Section of Wildfire Assessment Model

Content Analysis

The task analysis identified the skills and knowledge necessary to perform wildfire assessment. The skills and knowledge were then separated into the following objectives:

- prerequisite objectives
- terminal objectives
- enabling objectives.

The course content was defined by the terminal objectives. Content to be acquired prior to the course was defined by the prerequisite objectives. Content to be acquired subsequent to the course was defined by the enabling objectives. Prerequisite objectives and enabling objectives were used to provide background, review, or enrichment content beyond the essential course content.

The content was divided into three categories: wildfire assessment, wildfire behaviour, and wildfire suppression. Terminal objectives focused on wildfire assessment. Because wildfire behavior and wildfire suppression are crucial for understanding the assessment process, they were also included as part of the assessment course, but only as review and enrichment content. With the task analysis completed and the content specified, course design began.

Instructional Design

The goal of this project was to produce an interactive learning system that would provide realistic wildfire assessment training to students with diverse entry skills who required training in a variety of locations. Based on the content analysis, a tutorial to teach the wildfire assessment process and a wildfire simulation for practice and for evaluating mastery of terminal objectives were chosen. As knowledge of wildfire behaviour and wildfire suppression was necessary for understanding wildfire assessment, and as users entry skills would differ, a review of wildfire behaviour and suppression was ap-

propriate. However, the focus of the course design remained on the wildfire assessment tutorial and on the wildfire assessment simulation.

The tutorial and simulation were modelled on the simulation model developed in the task analysis. Each was detailed on a flowchart that described the steps in the process, the decision points, and the outcomes of each decision. In the tutorial, an expert firefighter, Norm, is observed as he goes about his daily routine. Users follow Norm from his home to his office, then to a real fire. They interact with him as he makes decisions about wildfire assessment.

The simulation was designed using a computer graphic fire growth model. This model graphically depicts a fire modified by variables including weather, fuel type, and topography. Changes in and combinations of these variables modify the fire depicted in the model. Users decide what suppressive action is required and react accordingly.

To get information on wildfire behaviour and wildfire suppression, a hyperbase was designed. Users access information from this hypertext database in two ways: by browsing through topic-related material and linking to other topics, or by linking directly from the tutorial to related topics within the hyperbase.

Authoring

Successful completion of the instructional design in the form of a flowchart requires considerable time and is one of the most difficult tasks in the design process. When the major elements of the flowchart are complete, storyboarding begins. The specific narrative and text elements are written and the video elements (both still and motion), the computer graphic elements, and the computer branching information are described. During this phase, it is particularly important that the author, instructional designer, director, graphics designer, systems designer, and subject matter expert work closely to ensure that the final storyboard is consistent with the instructional strategy, that the visuals and computer elements are compatible, and that the storyboard reflects the content as defined by the task analysis.

Production

The completed storyboard is the blue print for the final production. From this, the director, graphics designer, and interactive systems designer begin to integrate all elements of the production. The director gathers the video elements from stock or from scheduled shoots and supervises the final editing of a master tape. The graphics designer produces the graphic elements and prepares them for inclusion on the edit master -- if they are video graphics -- or gives them to the systems designer who includes them in the courseware -- if they are computer graphics. The interactive systems designer produces the supporting courseware using an authoring system or, where necessary, a lower level computer language to control specific applications.

After the final edit master is complete, a check disc is produced and used to field test a prototype system. The prototype is evaluated for its effectiveness in meeting the objectives of the project. Revisions are made, then the videodisc is mastered and replicates are made. The replicates are packaged and distributed by the marketing department at ACCESS NETWORK.

Evaluation

The formal evaluation is the final step in the design process. The purpose of the evaluation is twofold: to determine the effectiveness of the courseware and to determine the effectiveness of the design process. All projects have some form of final evaluation, but the extent of the evaluation varies from project to project. The formal evaluation scheduled for the end of the project will provide the ETC with useful information on the effectiveness of the wildfire assessment computer videodisc system and for improving future projects.

Conclusion

Developing a computer videodisc system to train forest fire control personnel is a complicated process that requires extensive planning to ensure a quality product. The collective expertise of a team of professionals is essential to produce an effective training product. We at the ETC believe that the process utilized in this production will produce an effective product -- a computer videodisc system to train wildfire fighters on initial assessment. Confirmation of this belief must wait until the final evaluation is complete. The video elements and computer programming are in production and a finished product is expected by January 1990.

LEARNING ENVIRONMENT/MILIEUX D'APPRENTISSAGE

LOGO TWENTY YEARS LATER: FROM A STILL AMBIGUOUS "THEORY" TO A STILL AMBIGUOUS "REALITY"

C. Lamontagne, M. Roberge, H. Boyer
School of Psychology, University of Ottawa
Ottawa, ON

LOGO needs no introductory comments in the field of educational technology in general and of microcomputer-based educational technology in particular. The theoretical rationale behind it, on the other hand, clearly calls for introductory comments. Such comments might have been difficult to provide at all prior to 1980, given the scarcity of elaborate written statements on the matter by LOGO's father. With the publication of Mindstorms (Papert, 1980) one was allowed to dig a little deeper into LOGO's foundations, where lied the realm of the notion of computer-based microworlds, which made possible some comments on LOGO's foundations and the somewhat more powerful assessment of the wild hopes and claims that had accompanied LOGO through its first decade. However, much of the foundational power of the notion of microworlds was transferred, in Mindstorms, onto the so-called Society of Mind "theory", as ultimate foundational instance, a "theory" which was still being developed at that time and could not be presented very clearly. This further need for clarification was only very partally solved by the publication, six years later, of The Society of Mind (Minsky 1986), which leads us to just about now, no significant additions having been made to the theory since then. After acknowledging the revolutionary nature and exceptional power of what seems clear in the LOGO project, we turn to digging out ambiguities in Society of Mind Theory itself, and in the relationship between Society of Mind Theory and some of Papert's more intuitive foundational comments, and discussing them in view of their possible bearing on some of those aspects of the project that can be argued to have gone wrong. The ambiguities relate to the concepts of procedurality, alogicality and modularity of thought processes involved in learning. They are called upon to account for the limited success of Papert's project in terms of the current disappointing lack of new microworlds, other than Turtle Geometry, and the disappointing partial failure to see LOGO move into the classroom as a "discovery oriented" pedagogical context.

LOGO VINGT ANS PLUS TARD : D'UNE "THÉORIE" ENCORE AMBIGUË À UNE "RÉALITÉ" ENCORE AMBIGUË

LOGO n'a pas besoin de présentation dans le domaine de la technologie éducative en général et dans le secteur de la technologie éducative pilotée par micro-ordinateur en particulier. La théorie qui le sous-tend, par contre, doit clairement être commentée. Celle-ci pouvait être difficile à formuler jusqu'en 1980, étant donné la rareté d'écrits sur la question par l'auteur de LOGO. Avec la publication de Mindstorms (Papert, 1980) on a pu examiner davantage les fondements de LOGO où "se cachait" la notion de micromondes automatisés, ce qui a permis de formuler certains commentaires sur les fondements de LOGO et de faire une évaluation beaucoup mieux appuyée des espoirs et des promesses extravagantes qu'on attribuait à LOGO tout au long de sa première décennie. Cependant, une bonne partie du pouvoir fondamental de la notion des micromondes a été transformée, dans Mindstorm, en ce que l'on a appelé la "théorie" de la Society of Mind, comme cause fondamentale ultime, théorie qui était encore en voie d'élaboration à cette époque et qu'on ne pouvait exposer de façon très nette. La nécessité de cette clarification n'a reçu qu'une réponse très partielle au moment de la publication, six ans plus tard, de l'ouvrage intitulé The Society of Mind (Minsky, 1986) ce qui nous amène presque à aujourd'hui, la théorie n'ayant pas été précisé de façon significative depuis lors. Reconnaissant la nature révolutionnaire et le pouvoir exceptionnel de ce qui semble clair dans le projet LOGO, nous relevons des ambiguïtés dans la théorie de la Society of Mind même et dans la relation entre cette théorie et certaines remarques fondamentales plus intuitives de Papert et nous les analysons par rapport à leurs incidences possibles sur certains des aspects du projet dont on peut dire qu'ils se sont révélés erronés. Les ambiguïtés ont trait aux notions de "procéduralité" "alogicalité" et "modularité" des opérations mentales qu'implique l'apprentissage. On en fait état pour justifier le succès limité du projet de Papert et notamment le désappointant manque de nouveaux micromondes, autre que la géométrie de la tortue, et le désapointant échec partiel de LOGO à s'imposer dans les salles de cours comme contexte pédagogique "orienté vers la découverte".

LOGO TWENTY YEARS LATER: FROM A STILL AMBIGUOUS "THEORY" TO A STILL AMBIGUOUS "REALITY"

C. Lamontagne, M. Roberge, H. Boyer

School of Psychology
University of Ottawa
Ottawa, ON

Although it is difficult to put an exact date on LOGO's birth, we can quite safely assert that it is now well over 18 years old, and has thus entered adulthood. Although the identity crisis is over, LOGO having established itself beyond any doubt in a great many people's minds as a theoretically sound and practically powerful pedagogical tool, the debate over the extent of its theoretical foundations and the exact nature of its pedagogical impact is, in our view, still wide open. These two issues, of the extent of LOGO's theoretical foundations and of the exact nature of its pedagogical impact, are of course highly inter-related: "Actual" impact only makes sense in the light of "intended" impact, and inasmuch as the intended impact remains unclear, so does any assessment of the actual impact. This means that LOGO's arguable "reality" as a pedagogical tool can only emerge from LOGO's intended pedagogical impact, as contained in LOGO's "theoretical foundations", and that, consequently, the extent of LOGO's arguable "reality" will be a direct function of the extent of its "theoretical foundations". Our purpose in the present paper is to sketch out what we view as being the essential foundational principles behind LOGO, attempting to make explicit their exact *extents* as founding principles.

As the purpose of this paper implies drawing people's attention to the *limitations* of LOGO's theoretical foundations, it is possible that the paper be viewed as opposing LOGO. It is of utmost importance to insist that not only is this not our intention, but that our purpose is the exact opposite. Our interest in clarifying the extent of LOGO's foundational support lies in our concern for *extending* LOGO and *creating* new microworlds in a Papertian spirit. It is the extraordinary richness of Papert's views that has motivated us to search for limitations that could be identified, for opening up yet new alleys for developing yet more exciting and powerful pedagogical tools. Papert used to like to quote W.S. McCulloch's rebuttal to some critics: "Don't bite my finger, look where I am pointing"[1]: Our work should be interpreted to be aimed at attempting to bring into sharper focus what characterizes the direction in which Papert's finger is currently pointing, so that movement forward can continue, and maybe accelerate. There is certainly no "biting" intention to our argument.

1. Sources

LOGO enthusiasts of the first days had to wait for some ten years before getting their "sourcebook" of Papertian pedagogy. It came in 1980, with the exotic title of *Mindstorms* (Papert, 1980)[2]. The book was very widely read, by researchers and school teachers alike, and did provide a wealth of foundational views and arguments, presented in a rather informal manner. One major foundational claim was that important aspects of the theory of the learning mind upon which LOGO was built had to do with a so-called Society-of-mind theory of cognition, which Papert was currently developing in close collaboration with Marvin Minsky. *Mindstorms* enthusiasts of the early eighties therefore turned their expectations towards this theory to come, which was to provide major additions to *Mindstorms* in terms of more formal theoretical foundations. In 1985, the long awaited book on Society-of-mind theory came out (Minsky, 1985)[3]. It was authored by Minsky alone but contained ample recognition of Papert's involvement in the genesis of the theory, containing for instance a chapter entitled "Papert's Principle", which was but a very slightly generalized version of Papert's "conservation" example of the ways of the Society-of-mind metaphor, as presented in *mindstorms*. Disappointingly, in our view, Minsky's book hardly contained more than Mindstorms did, in terms of relevant theoretical foundations for Papertian pedagogy, leaving us with *Mindstorms* as sole source of insight into the basic architecture of LOGO's theoretical foundations. This remains true in 1989, no significant update of the foundational discourse having (to our knowledge) been proposed lately. Our argument will therefore almost exclusively be based on *mindstorms*; so much so in fact that, hereafter, "Papert" will be taken to mean "Papert (1980)".

2. Argument

Repeated exposure to *Mindstorms* led us to conceive of Papert's foundational effort as extending over two embedded levels of preoccupation: a more global one, concerned with adopting a stance vis-à-vis the issue of the growth of knowledge, and a more local one, concerned with adopting a stance vis-à-vis the issue of how such minds as those governed by the adopted "growth of knowledge" principles can be argued to actually "function". We shall deal with each level separately, in subsections 2.1 and 2.2 respectively.

2.1 Papert's stance vis-à-vis the issue of the growth of knowledge

For over two thousand years at least, human kind has almost continuously been explicitly concerned with the mystery of the acquisition and growth of knowledge. The positions thinkers have adopted throughout all these centuries can metaphorically be compared to the positions a violinist's fingers can occupy on a string, to give rise to the variety of sounds that he wishes to produce; in the case of our thinkers, the anchor points, or poles, allowing the initial tension of the string are that of *naïve realism* at the low end, holding that the "knower" is but a (passive) "bucket" awaiting to be filled by absolutely "objective" knowledge, and that of *pure idealism* at the high end, holding that the "knower" is the sole absolute creator of whatever knowledge happens to emerge to consciousness, and holding that only the knower can be argued to exist. Almost every possible "note" has been played on that string at some point or other in the history of thought, but with

a progressive shift towards notes in the middle range, where an external reality is assumed to exist somewhat objectively, but where this external reality is only accessed indirectly by the "knower", who is taken to be incapable of self-exclusion in the process of knowing, therefore actively taking part in, and "subjectivising", the process. Now Jean Piaget's "interactionism" can be argued to fall within this so-called middle range of epistemological "musicality", based as it is on the view that knowledge schemes emerge out of one another (however mysteriously, through the so-called "process of reflective abstraction"[4]) under the constraining influence of intensive interaction with the environment. And Papert situates himself right at the heart of this Piagetian perspective, which we shall refer to, hereafter, as "*Genetic Interactionism*", "genetic" because it is essentially concerned with the genesis, or growth, of knowledge, and "interactionism" because it holds that knowledge can only be thought of as being the "product" of an "interaction" between subject and object, organism and environment.

Now adopting *Genetic Interactionism*, with the core tenet that "knowledge builds on knowledge", has implications which are worth making explicit. Firstly, *Genetic Interactionism* does not allow knowledge to have any kind of absolute ("mind independent") objectivity: current knowledge critically rests on chains of (however interactively driven) "reformulations" of previous knowledge states stretching back into the past as far as one can imagine (from personal history to specie history ...), and is therefore best apprehended through the study of its genesis. Secondly, *Genetic Interactionism* does not allow knowledge to have any kind of definitiveness to it, however "complete" it might appear, the next step up in the knowledge acquisition ladder critically depending upon its propensity to being repetitively reformulated into yet more powerful ways, chaining up into the future as far as one can imagine (from personal development to biological mutation ...). This pictures the evolving mind, at any point in time, as emerging from within the tension between its "history" and its "potentialities". We shall call "the learned mind" that aspect of every evolving mind having to do with knowledge as it currently provides adaptive ability, plunging its roots into the "epistemic history" of the particular mind, and we shall call "the learner's mind" that aspect of every evolving mind having to do with knowledge as it allows for reformulations to take place, stretching its purpose into the epistemic potentialities of the particular mind. Figure 1(a) presents a schematic summary of this picture.

These implications of *Genetic Interactionism*, in turn, have important implications on formulating the *pedagogical intention*. Most important of all is the implication that the pedagogical intention has to be formulated in terms of a difference between two minds rather than between a mind and some objective knowledge. The two minds in question are that of some target learner on the one hand, which we shall call "the *learner's* mind" (note the italics), whose state of knowledge is argued to "miss" something, and that of some ideal expert on the other hand, which we shall call "the *learned* mind" (note the italics), whose state of knowledge is argued to include it. But in order for the pedagogical intention to be expressed in terms that are consistent with what was said in the last paragraph about individual minds, it must be stressed that both the *learner's* mind and the *learned* mind are full-fledged minds in their own rights, each with both an internal "learned mind" aspect (note the plain characters) and an internal "learner's mind" aspect (note the plain characters). In the case of the *learner's* mind, of course, it is its "learner's mind" aspect which happens to be of particular

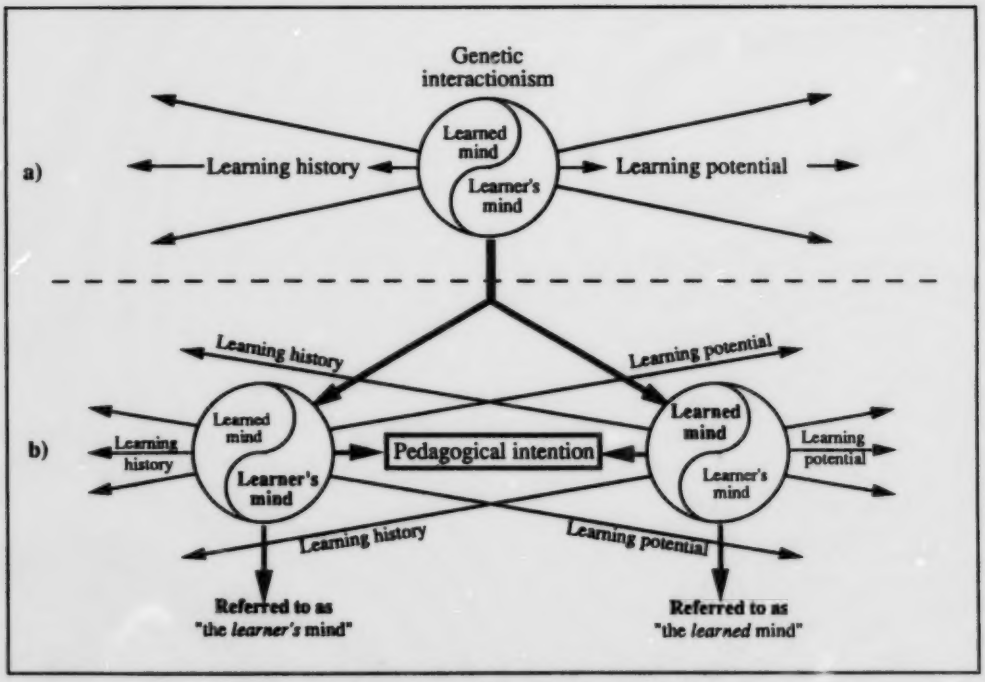


Figure 1. Implications of genetic interactionism on the nature of the evolving mind (a) and on the nature of the pedagogical intention (b).

significance and which must be spelled out in terms of how its *learning potentialities* can support the forward epistemic stretch towards the chosen *learned* mind, whereas in the case of the *learned* mind, it is its "learned mind" aspect which happens to be of particular significance and which must be spelled out in terms of how its *learning history* can support the backward epistemic stretch out towards the *learner's* mind. Thus, the idea is that an optimally formulated pedagogical intention should not only specify the respective knowledge states of the *learner's* mind and of the *learned* mind whose contrast sketches out the scope of the pedagogical task, but should do it in a way where the historical (or genetic) grounding of the given *learned* mind are explicitly shown to meet with the learning potentialities of the given *learner's* mind. Figure 1(b) presents a schematic summary of this way of viewing how *Genetic Interactionism* affects the formulation of the pedagogical intention.

As can be easily imagined, this opens up a Pandora's box for the applied researcher in Education: Not only does it imply finding theories of *learners'* and of *learned* minds, which is already quite a contract, but the theories in question also have to be *genetic* ones, which complicates matters almost to the point of hopelessness, ... which might explain why most researchers shy away from it, reverting to naïve realism, which is epistemologically somewhat less satisfactory, but which has the major advantage of not complicating matters too much: The "bucket" theory of knowledge acquisition calls for no particular theory of the *learner's* mind, apart from requiring some kind of "open"-mindedness on the part of the learner, not any more than it calls for a particular theory of the *learned* mind, knowledge being assumed to exist objectively, free of any particular mind's ruling. Papert is one of the very few who faced the challenge of bringing *genetic interactionism* all the way from the philosopher's armchair and into the classroom. Turning to addressing the way in which he dealt with the double implication of providing a genetic theory of the *learner's* mind and of providing a genetic theory of the *learned* mind brings us to what we referred to, in introducing Section 2, as being Papert's second level of preoccupation in its foundational effort. Let us then acknowledge this analytical distinction by moving to the next sub-section to deal with this so-called second level of preoccupation, that of the issue of how *learners'* minds and *learned* minds can be argued to actually "function."

2.2 Papert's stance vis-à-vis the issue of how *learners'* minds and *learned* minds actually "function"

Most of what Papert has to say about how *learners'* minds and *learned* minds actually function turns out to lie at a level of generality which makes it applicable to both types at once, although *Mindstorms* is structured in a way that makes the *learner's* mind overwhelmingly more central in the explicit discourse. Papert's theory of how minds function can be argued to be discussed at two different levels: a more macroscopic one, and a more microscopic one.

At the more macroscopic level, Papert argues for the "multi-dimensionality" of the mind, insisting on the necessity to call upon *emotional* and *cultural* dimensions, as well as upon the usual more purely *cognitive* dimensions, in elaborating a model of the functioning mind. The emotional dimension is mentioned here and there in relation with the *learner's* mind, but is most interestingly brought to the fore in *Mindstorm*'s Epilogue, devoted to one particular type of *learned* mind, that of the mathematician, as the role of *aesthetics* in mathematical "thinking" is discussed. As for the cultural dimension, it's importance is mostly stressed "en passant" as Papert discusses the role of cultural meaningfulness for the *learner's* mind, for example in connection with the "computer culture", and as he stresses how he departs from the canonical Piagetian attitude of making innate factors responsible for the sequencing of developmental stages, irrespective of cultural influences. Finally, the cognitive dimension receives "macroscopic" attention in two (arguably contradictory) ways, where on the one hand a case is made for the usefulness of *metacognition* for the *learner's* mind ("We can learn more, and more quickly, by taking conscious control of the learning process, articulating and analyzing our behavior."p.113), and where on the other hand a case is made for the usefulness of *unconscious* "processing" for the *learned* mind (as argued in *Mindstorm*'s already mentioned Epilogue, entitled "The Mathematical Unconscious"). Of course, what we have called here the "macroscopic" level of discourse can only be an introduction to Papert's theory of how the mind functions, an introduction which must open up onto a much more detailed discourse, where the ways of the mind get scrutinized at a much more microscopic level.

At the more microscopic level of discourse, then, we find one major reference: The Society-of-mind theory. Although it certainly does not exclude the *multi-dimensionality* principle, it is important to note right from the start that it is extremely difficult to see how the theory can actually be made to actively implement it, for either the *learner's* mind or the *learned* mind, but the theory does provide a view of the more purely cognitive aspects of the mind which appears quite relevant for modeling both kinds of minds.

Society-of-mind theory proposes to view minds as societies of computational *agents* , each one quite "simple-minded", but "socially" organized in ways that allow for quite complex cognitive activity. The key-features of the theory are *procedurality*, *modularity*, and *alogicality*.

Procedurality means that a Society-of-mind *agent* is an "effective decision procedure", as defined by the mathematical theory of computation. Any aspect of the functioning of any mind which cannot be expressed in procedural terms must therefore be denied the support of Society-of-mind. This can be considered somewhat problematical for Papert, for he does also mention that the ways of the mind include *non-procedural Cognition*, about which little else is said! In any case, coming back to procedural cognition, we find *procedurality* critically linked to *systematicity* and *debugging* (i.e. ridding thoughts of their inconsistencies). That Society-of-mind does allow for inconsistencies is clearly deducible from its *alogicality*, which is nothing but the acknowledgement that the ways in which *agents* are made to interact within the "Society" need in no way to be set according to the laws of logic. This presents yet another slight problem for Papert, who also sometimes argues in a way that might lead the reader to believe that there can be "bug-free" cognition (or *true theorizing*), presumably in some *learned* minds escaping the ruling of Society-of-mind's *alogicality*, setting limitations to the "debugging philosophy". In any case, for *debugging* to be carried out efficiently, *modularity* is required. *Modularity* , third and last main feature of the theory, is this ability Society-of-mind has of allowing *agents* (or *procedures*) to be defined in terms of units which can be used in almost as wide a variety of "societal" patterns, or "meanings", as desired. This ability of course takes on an immense appeal in the context of the *Genetic interactionist's* search for "dynamic" theories of knowledge, where knowledge is conceived of as being endlessly reformulatable into more and more encompassing forms, as well as being the product of such reformulations.

Given all this, then, how does Society-of-mind constitute the theory required by the applied researcher in Education in order to formulate clearly some given pedagogical intention, spelling out a detailed model of some *learner's* mind and a detailed model of some *learned* mind, fitting the latter's learning history into the former's learning potential, and creating a pedagogical tool along these lines? For what it does cover, Society-of-mind does constitute an original and powerful theory. But there certainly is much more to the required story than what Papert tells us. What is missing is quite obviously a clear picture of how **particular** groups of "alogical modular procedural agents" do account for **particular** cognitive "acts" which bear some relevance in the context of **particular** pedagogical intentions. Papert's Society-of-mind strikes us as being a *language* for formulating theories of cognition, much more than as being a *theory* of cognition as such: the language is presented, but Papert does not seem to use it to utter actual sentences ... to provide us with models of *actual cognition*! Even in the case of the regularly used simple example of the FD and RT Turtle Geometry primitives, it is difficult to conceive of Society-of-mind scenarios which would document Papert's claim to "syntonic" relevance. This weakness can be most strongly sensed in the extension of the concept of *modularity* into the concept of "mind-sizedness", where Papert talks of the necessity for the learner to deal with "mind-size bites" of knowledge. *Modularity* is one thing, "mind-size modules" of knowledge is another one, for the latter implies very specific modules, modules which "fit" a particular pedagogical intention, which can be "chewed" by the *learner* while "tasting" the *learned*: How does one decide on what a "mind-size bite" is, in the absence of a clear theory of the particular *learner's* mind and of the particular *learned* mind whose tension give rise to the particular pedagogical intention?

But LOGO stands, as an obvious mouth-watering banquet of actual and potential "mind-size bites". How did Papert manage it? As should be clear from what was said above, the answer cannot lie in what he has managed to make explicit in *Mindstorms*! In our view, the answer lies deep within Papert's own *private* Society-of-mind, with those *agents*, specialized in *learned* minds, which we might call "Papert-the-intuitive-genetic-mathematician" or "Papert-the-intuitive-genetic-physicist", and those *agents*, specialized in *learner's* minds, which we might call "Papert-the-intuitive-genetic-psychologist" or "Papert-the-intuitive-genetic-anthropologist". But since they remain hidden, they can be of no help to the applied researcher in Education who does not himself enjoy such an intuitive wealth of pedagogically relevant inner societal dynamics. Thus, Microworld design remains the exclusive prerogative of those exceptional minds which can make up for the formidable holes left in Papert's explicit rationale, and will remain so until what allows them to do it is made explicit, thereby allowing everyone to join in and start exploiting the full potential of the paradigm.

3. References

[1] McCulloch, W.S. (1965). *Embodiments of Mind*. Cambridge, MA: MIT Press.

[2] Papert, S. (1980). *Mindstorms*. NewYork: Basic Books.

[3] Minsky, M. (1985). *The Society of Mind*. New York: Simon and Schuster.

[4] See Lamontagne, C. (1987). Sensorimotor emergence: Proposing a computational "syntax". In W. Callebaut and R. Pinxten (eds.), *Evolutionary Epistemology*. Reidel.

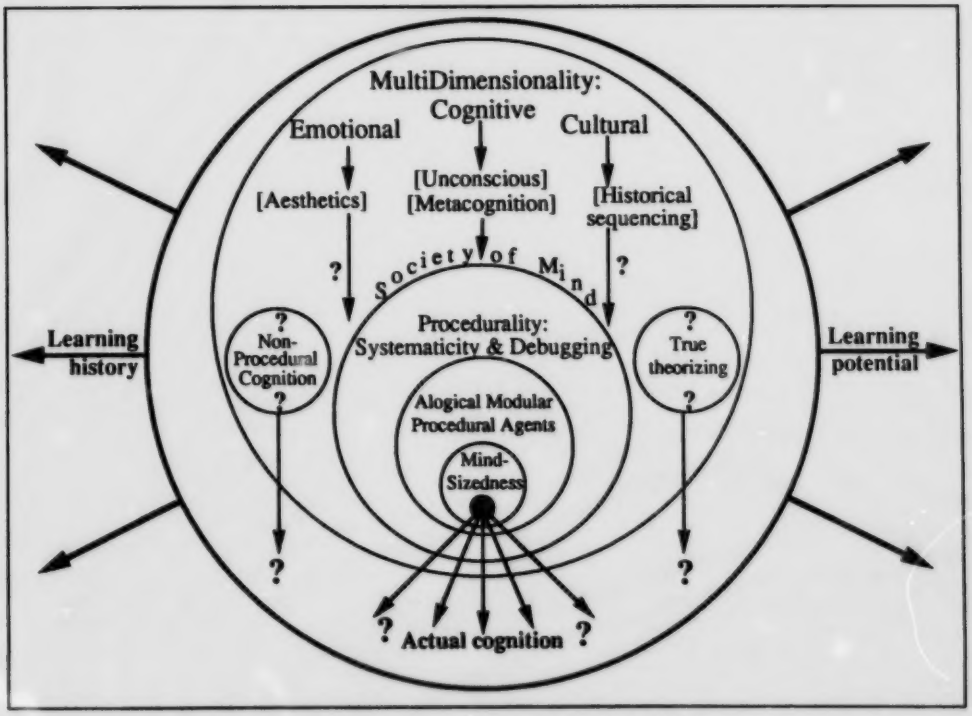


Figure 2. Summary of Papert's more detailed stance with respect to the nature of the mind.

UN ENVIRONNEMENT OBJET D'EXPLORATION D'APPRENTISSAGE

I. Borne
École des hauts études en informatique
Université René Descartes
Paris, France

C. Girardot
École normale d'Auteuil
Paris, France

Les techniques de représentation des connaissances en intelligence artificielle sont de plus en plus utilisées pour concevoir des applications éducatives. Or, la principale caractéristique d'un langage objet est justement de permettre une correspondance aisée entre un objet du monde réel et un objet informatique. L'environnement Smalltalk-80 est trop complexe et mal adapté pour être livré tel quel à des novices. Contrairement aux langages classiques tel que Logo il n'existe pas à notre connaissance de travaux sur les apports éventuels en milieu scolaire de la programmation par objet. Nous avons donc développé, au dessus de Smalltalk-80, un environnement d'exploration éducative dans un double but: d'une part pour explorer des contenus pédagogiques et d'autre part à travers ceux-ci faire découvrir et utiliser les concepts d'objet et de classe. Notre environnement est indépendant du domaine d'étude choisi, il a été notamment utilisé pour l'étude des transformations géométriques mais aussi la mesure de surfaces, la conjugaison des verbes, les objets musicaux , les formes, les couleurs, etc. Par ailleurs il est construit suivant des principes classiques, i.e. l'écran doit être la représentation fidèle de la perception de l'utilisateur, et exploite la représentation multiple des objets.

Une première expérimentation avec des instituteurs a permis de vérifier l'efficacité et l'adéquation de notre environnement. D'un point de vue pédagogique il apparaît que ce type d'environnement de travail, riche et ouvert, offre des perspectives intéressantes notamment pour la formulation d'hypothèses, l'élaboration de stratégies et la construction de concepts fondamentaux par raffinements successifs, perspectives qu'il reste à confirmer par un travail d'expérimentation à long terme et de réflexion.

OBJET ORIENTED ENVIRONMENT FOR LEARNING EXPLORATION

Knowledge representtion techniques in artificial intelligence are more and more used to design educational applications. The main feature of an object oriented language is precisely to allow easy correspondence between a real world object and a computational object. The Smalltalk-80 environment is too complex and poorly suited for novices. Contrary to classical languages such as Logo, we do not know what works in schools about possible contributions of the object oriented programming approach, so we have developed, in Smalltalk, an exploration environment aiming at two goals: on the one hand to explore curricula and on the other hand, through them, to discover and to use object and class concepts. Our environment is independent from the study domain. It was used, in particular, in elementary school to study geometrical transformations but also for measurement of surfaces, conjugation of verbs, musical objects, forms, colors, etc. Furthermore, its design has followed widespread principles: "what you see is what you get" and it exploits multiple representations of objects.

Through experimentation with teachers, the effectiveness and adaptation has been tested. At the pedagogical level this type of working environment appears to have considerable potential, especially in hypothesis formulation, strategies elaboration and in fundamental concepts building by successive refinement. This remains to be further developed through additional theoretical work and long-term experimentation.

UN ENVIRONNEMENT OBJET D'EXPLORATION D'APPRENTISSAGE

Isabelle BORNE* Colette GIRARDOT**

*EHEI - Université René Descartes, **Ecole Normale d'Auteuil
Paris, France

1- Introduction

Les langages objets ont su se faire une place dans le domaine de l'intelligence artificielle. Leur principale caractéristique est de permettre une correspondance naturelle entre un objet du monde réel et un objet informatique, c'est une des raisons de leur succès, en particulier auprès des programmeurs novices. Plus encore nous pensons que la programmation par objet a sa place à l'école. Jusqu'à présent, les expériences d'introduction de la programmation à l'école élémentaire ont été faites avec le langage Logo. Mais en Logo, par exemple, s'il est facile de dessiner un carré, il est plus difficile de penser le carré comme un objet auquel on demande de se dessiner ou de changer ses proportions[2]. Logo n'est-il pas le langage qui a eu le plus grand succès en éducation parce qu'il possède un objet pédagogique formidable : la tortue ? A ce titre on peut qualifier Logo de langage mono-objet.

Smalltalk est non seulement considéré comme le pionnier de la grande famille des langages objets qui existent à l'heure actuelle mais c'est aussi le système le plus uniforme car il est bâti sur le seul paradigme de l'objet. A l'origine, le projet Smalltalk avait pour but de créer de nouvelles technologies pour à la fois enseigner la programmation et réaliser des activités éducatives. Les expériences d'Adele Goldberg effectuées avec le système Smalltalk-72 entre 1974 et 1976, avec des enfants de douze et treize ans, ont permis de montrer que ce langage est enseignable[6] dès cet âge. Les développements qui ont suivi ont abouti au système Smalltalk-80, plus spécialement destiné aux professionnels de l'informatique[7]. C'est un outil volumineux, puissant et largement répandu, qu'il faut considérer comme une grande bibliothèque d'objets dans laquelle puiser pour construire son propre environnement. C'est pourquoi, malgré nos premiers travaux sur la conception d'un petit langage objet destiné à l'école[3], nous avons décidé de nous appuyer sur un vrai système professionnel, Smalltalk-80, pour bâtir un environnement de programmation objet utilisable à l'école par les enfants et les enseignants.

2- Les objectifs de départ et la démarche pédagogique

L'informatique est de plus en plus introduite à l'école et généralement sous deux formes différentes : soit comme support d'enseignement d'une autre discipline, soit en tant que discipline scientifique propre. La première concerne la résolution de problèmes prise dans son sens le plus large, la deuxième comprend en particulier l'enseignement de langages de programmation. La programmation n'est pas un but en soi, c'est un véhicule de transmission de concepts[13]. Notre approche regroupe ces deux aspects dans le but d'initier les enfants à la programmation par objet par l'intermédiaire d'exploration de contenus pédagogiques touchant le plus de disciplines possible (mathématiques, français, physique, musique etc.). Cette activité doit s'intégrer au programme scolaire, ce qui signifie que le logiciel doit être disponible sur micro-ordinateur et permettre un dialogue en français. Elle doit également amener progressivement les enfants à découvrir les concepts d'objet, de message, de classe et d'héritage.

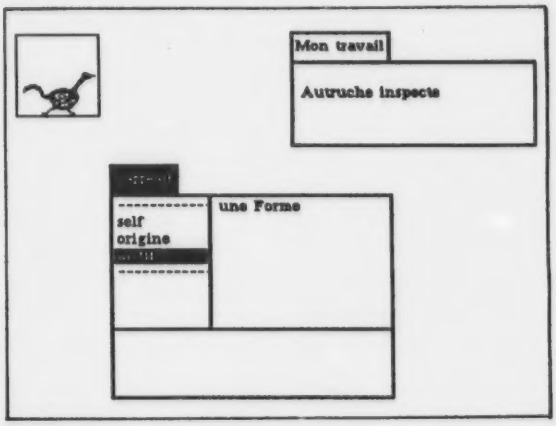
Par ailleurs, la programmation par objet est de plus en plus enseignée dans les milieux professionnels où elle se prête bien au prototypage d'applications, elle pose cependant plusieurs problèmes. Dans le cas de Smalltalk-80, la compréhension des concepts de base et la manipulation de l'environnement sont très vite assimilées, les difficultés arrivent au moment d'appliquer ces concepts car les gens ont déjà pris des habitudes de programmation avec un langage procédural. Il ressort qu'il est plus facile d'enseigner la programmation par objet à des apprentis programmeurs qu'à des professionnels auxquels on demande de changer leurs habitudes[11].

L'apprentissage de la programmation est typiquement un domaine de recherche

pluridisciplinaire. Notre recherche regroupe des spécialistes de l'informatique, de l'éducation et de la psychologie, en particulier en didactique de l'informatique. Elle s'intéresse aux problèmes d'apprentissage de la programmation et au développement de méthodes pédagogiques d'enseignement de la programmation par objet comme cela a été fait pour d'autres langages[10]. Apprendre à programmer pose un certain nombre de problèmes intéressants[12] car il ne s'agit pas simplement d'apprendre la syntaxe d'un langage, mais aussi d'apprendre à formaliser et à résoudre un problème concret. Cela implique plusieurs tâches comme analyser le problème à résoudre, le décomposer en sous-problèmes, raisonner par analogie, généraliser certains concepts, corriger les erreurs. L'approche pédagogique proposée par Adele Goldberg[6] est celle que l'on adopte intuitivement. Elle consiste à proposer au départ un exemple complètement réalisé , c'est-à-dire une classe et des instances de cette classe. Puis, on propose à l'élève d'observer les objets et leurs comportements en leur envoyant des messages, en les modifiant, en créant de nouveaux objets et de nouveaux messages. La création de nouvelles classes, qui demande un niveau d'abstraction supplémentaire, sera introduite plus tard ainsi que le mécanisme d'héritage pour créer des sous-classes.

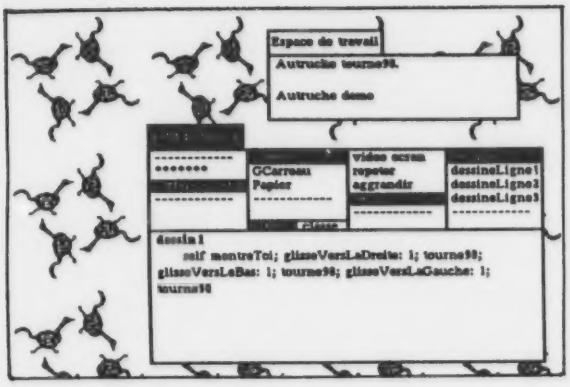
3- L'environnement objet

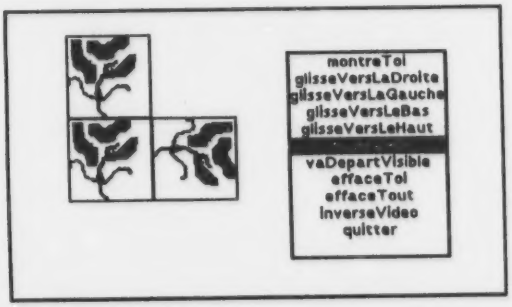
Nous avons développé au dessus de Smalltalk-80 une couche représentant un nouvel environnement qui donne, aux enfants et aux enseignants, accès à la pleine puissance de Smalltalk-80. Il se présente comme un système Smalltalk français (toutes les parties visibles du système ont été mises en français) dans lequel nous avons caché les classes du système qui ne nous intéressaient pas, redéfini en français certaines autres et simplifié leur utilisation si nécessaire (nombres, crayon, rectangle, rond ...), gardé en les adaptant certains outils et ajouté des facilités notamment pour définir les classes, les méthodes et les objets complexes. Cet environnement a donc comme principe de ne montrer que ce qui est nécessaire et les activités qui seront développées le seront suivant ce principe, c'est-à-dire qu'elles ne présenteront à l'élève que les objets et les classes utiles à son travail. Cependant, il sera toujours possible d'élargir le système avec des classes du système modifiées ou redéfinies en français et situées dans des fichiers, ou avec des classes d'exemples développées pour d'autres activités. Parmi les outils modifiés, signalons l'inspecteur qui permet d'explorer un objet et de modifier ses caractéristiques. Nous avons voulu qu'il soit l'outil privilégié de présentation des différents aspects d'un objet, ainsi si l'objet possède une représentation graphique ou textuelle on pourra le voir, s'il possède une représentation sonore on pourra l'entendre et on pourra suivre son évolution.



4- Expérimentation

L'introduction de la programmation par objet devant se situer dans le contexte scolaire normal, ce sont aux éducateurs et non aux chercheurs de définir les activités qui seront faites en classe et d'encadrer les enfants. Les instituteurs doivent donc maîtriser parfaitement l'outil informatique et préparer des séances de travail pour les enfants. Nous avons formé une dizaine d'instituteurs avec lesquels nous avons pu valider en partie notre recherche et adapter encore mieux l'environnement à leurs besoins. Avec eux, nous avons constitué une petite bibliothèque d'activités éducatives dans différents domaines: petite séquence musicale, manipulation de formes graphiques, construction de papiers-peints, mesure de surface, conjugaisons de verbes. Après cette phase de formation des instituteurs, l'expérimentation avec les enfants devrait démarrer au troisième trimestre de cette année. Pour commencer nous travaillerons avec les deux dernières classes de l'école élémentaire, c'est-à-dire des enfants de neuf et dix ans, pour lesquels nous avons écrit un programme en Smalltalk fondé sur la découverte des transformations géométriques par simulation et création de papiers peints[5]. Cet exemple montre la puissance de modélisation, d'interaction et d'animation offerte par la programmation par objet.





Les enfants se familiariseront avec les univers proposés puis pourront les enrichir en créant eux-mêmes d'autres méthodes puis de nouvelles classes. Les enfants exprimeront sûrement des besoins d'objets que nous n'aurons pas prévus et que nous devrons à ce moment-là mettre à leur disposition.

5- Conclusion

Notre but étant de faire effectivement programmer les enfants avec les concepts de la programmation par objet, notre démarche est différente de celles adoptées dans les travaux précédents[4,8]. Par ailleurs, il s'agit d'une expérience nouvelle car, à part le rapport d'Adele Goldberg, il n'existe pas à notre connaissance d'autres expérimentations en milieu scolaire avec un langage objet. Dans un premier temps, nous voulons mener à bout cette expérimentation à l'école, car les réactions des enfants sont essentielles et peuvent orienter, par la suite, le travail dans une direction plus privilégiée. Nous essaierons de comparer nos résultats avec ceux d'autres recherches, mais l'apprentissage de la programmation peut prendre des formes très variées suivant le système et l'environnement utilisés, les contraintes imposées par l'enseignant, le type de problème posé et le temps consacré à programmer. Contrôler une expérience demande donc de prendre en compte un grand nombre de paramètres et des comparaisons de résultats ne sont possible que si les mêmes paramètres ont été pris en compte.

Notre effort va se poursuivre au niveau de l'interface notamment pour faciliter la création d'applications interactives qui passe par la manipulation d'objets complexes. Pour cela il nous faut exploiter à fond le style de la programmation par objet en proposant de nouveaux outils pilotés par des menus et ajouter une composante de construction automatique de programme[9]. Cette première expérimentation concerne l'école élémentaire, mais notre environnement pourra être utilisé pour l'introduction de Smalltalk dans les lycées dans le cadre de l'option informatique sur la région parisienne[1]. L'environnement objet permet de travailler sur l'apprentissage de concepts plus spécifiques à des âges variés en jouant sur la complexité croissante des objets bâtis à partir d'objets précédfinis et en réglant le niveau d'ouverture sur le système Smalltalk-80.

Références

1 - Georges-Louis Baron, "La naissance de l'informatique comme discipline scolaire de formation générale", AFCET/Interfaces, n° 69, pages 3-15, Juillet 1988.

2 - Isabelle Borne, "Les Objets dans Logo", Journée d'étude sur les Langages Objets, Le Cap d'Agde, BIGRE n° 37, 1983.

3- Isabelle Borne "Interprètes pour l'Ecole: Logo, Micro-Smalltalk", Thèse de 3ème cycle, Université Paris VI, rapport LITP 85-1, octobre 1984.

4 - Andrea diSessa, Harold Abelson, "Boxer: A Reconstructible Computational Medium", Communications of the ACM, Volume 29 Number 9, September 1986.

5 - Colette Girardot "Compte-rendu du travail sur Smalltalk", rapport interne présenté à la direction des écoles, Ecole Normale d'Auteuil, mai 1987.

6 - Adele Goldberg, Alan Kay, "Teaching Smalltalk", Xerox report SSL 77-2, June 1977.

7 - Adele Goldberg, Joan Ross, "Is the Smalltalk-80 System for Children ?", Byte, Volume 6 Number 8, August 1981.

8 - Laura Gould, William Finzer," Programming By Rehearsal", Technical report SCL-84-1, Xerox Palo Alto Research Center, May 1984.

9- Henry Lieberman, "Constructing Graphical User Interfaces by Example", Graphics Interface Conference, Toronto, Canada, May 1982.

10 - Janine Rogalski, "Enseignement de méthodes de programmation dans l'initiation à l'informatique", Colloque francophone sur la didactique de l'informatique, Université René Descartes, Paris 1-3 septembre1988.

11 - Dan Shafer, "Teaching OOP", Panel in OOPSLA'88 Proceedings, San Diego, California, September 25-30, 1988.

12 - Elliot Soloway, "Learning to program - learning to construct mechanisms and explanations", Communications of the ACM, Volume 29 Number 9, September 1986.

13 - Harald Wertz, "Some ideas on the Educational Use of Computers", Proceedings of the Annual Conference of the ACM, Los Angeles, November 9-11 1981.

THE COMPUTER-ASSISTED MEASUREMENT OF MUSICAL PERFORMANCE

F. St-Pierre
Dept. of Measurement, Evaluation and Computer Applications
Ontario Institute for Studies in Education
Toronto, ON

Findings from phonemic theory state that for students of second languages, interference is caused by sounds occurring in the target language but not in the speaker's mother tongue. These sounds interfere with the speaker's production and auditory discrimination abilities. In music performance as in speech production, performers constantly monitor their auditory feedback and adjust their output based on their knowledge of acceptable performance. This creates a problem during rehearsals for neophytes and experienced musicians learning a new musical style because they may not recognize when their production is appropriate or inappropriate; which often leads to the reinforcement of inappropriate technique. This investigation addresses the problem of producing a learning environment that would facilitate the process of a musician self-monitoring certain musical aspects of his or her performance. The focus of this study is restricted to the computer-assisted measurement of certain aspects of rhythmic interpretation and articulation of the musical phrase in the jazz idiom. The data are collected from subjects (undergraduate music students) using a Yamaha DX7 synthesizer (velocity sensitive with a weighted action) connected to a Macintosh micro-computer using the Musical Instrument Digital Interface (M.I.D.I.) protocol. The musical aspects of intensity and time are the focus of this study. The specific aspects of intensity monitored in this study are the degree of loudness (absolute intensity) and dynamic modulation (attack). Those aspects of time that are examined are the actual duration of tones and pauses, the time of attack and release, and certain artistic variations such as legatoness.

MESURE ASSISTÉE PAR ORDINATEUR DE LA PERFORMANCE MUSICALE

Les découvertes de la théorie des phonèmes indiquent que pour les étudiants de langue seconde, l'interférence provient de sons émis dans la langue cible, mais non pas dans la langue maternelle de la personne qui parle. Ces sons interfèrent avec les aptitudes de production et de discrimination auditive de la personne qui parle. Pendant une représentation musicale, tout comme pendant un discours, les exécutants surveillent continuellement les réactions de leur auditoire et ajustent leur tir en fonction de ce qu'ils savent être une performance acceptable. Pendant les répétitions, cela crée un problème pour les néophytes et les musiciens expérimentés qui apprennent un nouveau style musical, parce qu'il se peut qu'ils n'arrivent pas à déterminer si leur exécution est appropriée ou non, ce qui donne souvent lieu au renforcement d'une technique inappropriée. Le présente recherche traite du problème de la création d'une milieu d'apprentissage qui faciliterait la tâche du musicien qui doit auto-superviser certains aspects musicaux de sa performance. Le point central de cette étude ne porte que sur la mesure assistée par ordinateur de certains aspects de l'interprétation rythmique et de l'articulation de la phrase musicale de l'idiome du jazz. Les données sont recueillies auprès de sujets (étudiants au baccalauréat en musique) à l'aide d'un synthétiseur Yamaha DX7 (sensible à la vélocité avec action pondérée) relié à un micro-ordinateur Macintosh et piloté par le protocole Musical Instrument Digital Interface (M.I.D.I.). Les aspects musicaux d'intensité et de temps constituent le noyau de cette étude. Les aspects particuliers de l'intensité mesurés dans cette étude sont le volume (intensité absolue) et la modulation dynamique (attaque). Les aspects temporels visés sont la durée réelle des sons et des silences, le temps d'attaque et de relaxation et certaines variations artistiques comme le fait de jouer legato.

THE COMPUTER-ASSISTED MEASUREMENT OF MUSICAL PERFORMANCE

François St-Pierre

Ontario Institute for Studies in Education
Department of Measurement, Evaluation and Computer Applications
252 Bloor St. W., Toronto, Ontario

INTRODUCTION

In music performance as in speech production, performers constantly monitor their auditory feedback and adjust their output based on their knowledge of acceptable performance (Gabrielson, 1985; Gates & Bradshaw, 1974; Heller, 1969; Politzer, 1965; Sloboda, 1982). Neophytes and experienced musicians therefore may reinforce musical behavior inappropriate to the new idiom when musical events are introduced that are not part of their system of auditory discrimination (Coker, 1962; Newmann, 1982). This occurs because they may not recognize when their production is appropriate or inappropriate. An environment that would facilitate the process of a musician self-monitoring certain musical aspects of his or her performance and provide feedback as to the appropriateness of the performance may remedy this problem. Along these lines, Carl Seashore (1938) suggested that "*...to facilitate the acquisition of musical skills, objective instrumental aids may be used to great advantage, for both economy of time and precision of achievement. Among such aids are visual projection or quantitative indication of pitch, intensity, time, and timbre at the moment the tone is produced*". Feedback relating to other aspects of performance has also proven itself. For instatnce, visual electromyographic biofeedback helped wind players to control specific parts of the buccinator muscle in the cheek. (Sloboda, 1982).

This paper describes the first phase of development of an environment focusing on the measurement of certain aspects of rhythmic interpretation and articulation of the musical phrase in the jazz idiom. The first section describes previous research undertaken in this area. The second section contains the description of a functional prototype. The third and fourth sections describe the musical parameters chosen for study and the approach to their analysis. The last section describes areas of future research and users' reactions to small scale pilot testing of the prototype.

ANALYSIS BY MEASUREMENT

The approach to analysis by measurement of a musical performance may best be understood by contrasting it the *analysis by synthesis* approach. Researchers using the synthesis approach choose the musical criteria to be studied, and then synthesize (or digitize) multiple versions of musical examples of these criteria. All versions are slightly altered from one another, each having some systematic difference in terms of the performance variables (Bengsston and Gabrielson, 1982; Sundberg, Frydén & Askenfelt, 1983). By contrast, analysis by measurement attempts to describe (rather than manipulate) any or all of the four musical elements that may be under a performer's control, depending on the instrument. This approach is described by Seashore: "*All that is conveyed from the musician to the listener as music, is conveyed on sound waves... The sound waves are measurable, and there are only four variables which have musical significance: frequency, intensity, duration, and form...The psychological equivalents or correlates of these characteristics of sound are pitch, loudness, time and timbre. Rhythm, harmony, volume and tone quality are compounds of these; thought, feeling, action, memory, and imagination are in terms of these.*" (Seashore, 1938).

Innovative use of various technologies in developing environments for the acquisition and processing of musical sounds, is evidenced in this chronological sampling of previous studies. The Iowa Piano Camera (Henderson, Tiffin and Seashore 1936), is a modified acoustic piano that measured the pitch, key velocity, onset and release times and duration of notes with the use beams of light passing through the key mechanisms of the piano and projected onto a roll of film. This piano was used to study the musical aspects of intensity and time in piano playing. Small (1936) studied pitch, intensity and duration in pre-recorded (via phonograph) and live violin performances; pitch was measured by strobophotography, intensity by vacuum tube voltmeter, and duration by an electrically driven tuning fork. He attempted to describe aspects of violin playing that included under-holding and over-holding of notes, and the relationship between temporal deviation, intensity and melodic phrase structure. With the intent of developing new data acquisition techniques, Papich and Rainbow (1975) used contact microphones placed on stringed instruments to make multi-track recordings, which were then transformed into square wave forms by filters and a pitch discriminator. They then applied these technique to the study of intonation and pitch-vibrato of stringed instruments. Longuet-Higgins (1976) examined the conditions necessary for listeners to judge whether the onset of a note occurs "during" the current metrical unit rather than "at" its beginning or its end. They used a 48 note organ connected to a high speed paper punch. The generated information consisted of the key number and the time (in centiseconds) of each note's onset and release. Wills *et al* (1983) measured pedalling and fingering technique by devising a method which consisted of outfitting the pedal and each key of an acoustic grand with infra-red position sensors. The velocity and distance of key displacement were also recorded with a technique using infra-red light. Vercoe (1984) used a combination of a PDP-11/55, a real-time audio processor, filters, and a series of optical sensors installed on the keys of a flute to examine aspects of tempo. Rose (1985) examined the ratio of the tied triplet figure by using a spectrum analyzer and an Apple II+ to digitally code excerpts selected from recordings of professional jazz musicians. Pressing (1987) modified a DX7 synthesizer (to improve the temporal resolution offered by MIDI) to study musical parameters of improvised keyboard playing.

THE PROTOTYPE ENVIRONMENT

The current environment makes use of the Musical Instruments Digitial Interface (MIDI) protocol, a synthesizer which supports this protocol (including the MIDI Note Velocity parameter), and software routines for capturing and analyzing data. These routines are currently implemented on the Macintosh microcomputer.

In terms of ease of development and application, the principal advantages of using a MIDI-synthesizer based environment over a specially prepared acoustic instrument are a) availability, b) ease of use, c) portability, and d) low expense. (Naturally, in certain cases, a performer's expectations and needs may not be met by current synthesizer technology and so prepared acoustic instruments may be preferred.) Certain advantages of using MIDI over other methods of measurement such as digitization or transcriptions into standard musical notation, is the ease with which a note's velocity, onset-time and duration can be established. Gabrielson (1985) noted certain problems in determining precise note onset times for instruments with variable intonation with his technique (i.e. flute, bowed instruments, etc.) In the present research, the principal limitation of MIDI as a tool is data transfer rate (see Moore, 1987 for a description of the limitations of MIDI).

The basic i/o routines used in the program are necessary to capture musical information from the subjects as well as to provide an accompaniment. The driver routines are based on those described by Austin (1984). The data is captured in the form of a stream of MIDI bytes. Each byte received by the serial communications controller chip generates an interrupt which calls a routine that time stamps and stores the byte in a queue. The timing resolution is currently 1/1000th of a second. These time stamped bytes are fetched from this queue and parsed by a routine which attempts to attribute a *duration* for each note and store it in a record along with other information such as its velocity, onset time, and pitch etc. Twelve bars of MIDI information describing a walking bass line (which outlines the chord changes and serves to keep time for the subjects) has been hard-coded into the program and gets repeated at the top of every form. A slightly greater intensity (85 versus 64) is applied to beats 2 and 4 (within each bar) of the bass line, but the pitch adjusts itself if a change of key is requested by the performer.

MUSICAL PARAMETERS

The current system measures two musical parameters, intensity and time. The specific musical aspect of **intensity** monitored is that of dynamic modulation (attack), as measured by key velocity. The range of possible velocities for a sounding note is from 1 to 127 (a value of 0 turns a note off). According to the MIDI specification, a Velocity byte with a value of 64 (decimal) corresponds to ***mf***, 127 corresponds to ***fff***, and ***ppp*** corresponds to a value of 1. The musical aspects of time are the actual durations of tones and pauses, and the time of attack and release. **Onset** time is measured in absolute "pulses per quarter note" from the beginning of the recording, where each quarter note is divided into 300 pulses. **Duration** is derived by subtracting the onset time from the offset time (i.e. release of the note). As mentioned in the preceding section, all of this information is available immediately after a note (and its corresponding Note Off) is fetched from the input buffer.

ANALYSES

The measures listed in the preceding paragraph allow us to describe the incoming data in unambiguous terms (i.e. individual notes with various attributes). In order to interpret the program's output in terms of specific rules governing a jazz performance, the notes should be grouped into their phrases, or at least into series of notes. For instance, the following rules, drawn from the literature (Guiffre, 1969; Rizzo,1973) require that a unit of analysis smaller than or equal to a musical phrase be used:

- Rushing is caused within 8th note phrases by an improper relationship of 8th notes.
- the highest and last notes of a series receives an accent.
- Ghosted notes are always the lower notes in a series.

The importance of the musical phrase as a unit of organization and measurement has been acknowledged by many researchers (Henderson, 1981; Henderson, 1936; Longuet-Higgins, 1976; Simon and Sumner, 1968; Stein, 1962, Wessel et al, 1987). Its function can be described in familiar terms using human speech as a parallel: "*To a certain extent, phrasing is comparable to the delivery of speech. The speaker groups and shades the accentuations of the words according to sense; similarly the musical performer groups and diversely accentuates the notes that form musical shapes.*" (Stein, 1962). Musical devices for phrase definition include rests, harmonic progressions, articulation (i.e. the exact hitting of a note), temporal fluctuations, intensity etc. The use of rests as devices for the delineation of phrases was explored in this program using a two pass approach. On the first pass, tones belonging to chords are tagged, and values for rests are established. For the purposes of the present analyses, a rest is created between two tones whose inter-onset time is greater than a pre-established threshold value. The choice of these values is based on Al Henderson's analysis of Coleman Hawkins' use of rests at phrase endings Henderson (1981). During the second pass, the first and last notes of phrases and the top and bottom notes of chords are joined. In addition, the highest and lowest notes of each series of notes is determined and flagged, and future studies will attempt to describe the differences between novices and experts.

As shown in the figure below, two different phrases have been identified, distinguished by different hatch patterns for the lines. The dotted hatching represents phrases demarcated by rests whose value is equal to an eighth note triplet, those with solid lines by rests equal in value to a quarter note. The thick vertical lines join chordal tones (in the two bars at the right hand side), while the thin vertical lines represent "bar lines". The relative intensities of notes are not visible in the reproduction below, but the intensity of gray corresponds to the intensity (velocity) of the note in the original.

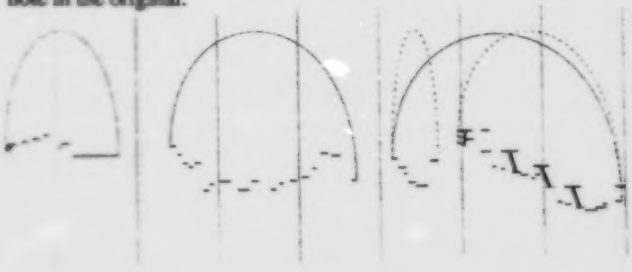


Figure 1. Phrases demarcated by the program

FIELD-TRIALS

The purpose of pilot testing was twofold: to obtain users' impressions of working within an environment of this sort as well as impressions, suggestions and criticism of this system in particular. The second purpose of pilot testing was to obtain some actual performance data from experts and novices for exploratory analysis and further tuning of the system. The environment was field-tested with 12 undergraduate music majors, three novice jazz musicians, and one professional musician. The participants were asked to play a "blues" in F (see Dannenburg and Mont-Reynaud, 1987; Coker, 1964) at a tempo of 200 beats per minute. The "feel" of the bass line andthe tempo, as well as various settings for the synthesizer were tested by a professional jazz musician prior to pilot testing. Unfamiliarity with the synthesizer keyboard was mentioned by a few as negatively affecting their playing, and two performers in the novice category were unable to follow the chord changes outlined by the bass line. A few suggested that the accompaniment should be played using a different tone than simply using the bottom end of the synthesizer they were performing on. As can be expected, not all phrases "found" by the system correspond to "musical" phrases, and so validation in the form of listening is required.The reactions to the environment in general were positive.

RESEARCH PLANS

The environment is currently at the end of the preliminary phase of development: data describing certain musical performance parameters can be collected from a live performance, this data can be parsed into meaningful chunks of information, and cursor feedback can be made available to the performer. Subsequent development will focus on two primary areas:

1) Two structures are currently being used to store the data from the performance: one structure is used for the incoming stream of MIDI bytes, while a second structure contains a higher level version of the analysis. This approach allowed the rapid development of a working prototype. However, a consolidation of these two data structures, as described by Free (1987) will facilitate the cross-referencing of *performance* and *analysis* data, which is not currently available. This is necessary in order to allow the system to be "tuned" interactively to many different parameters (i.e. values for rests, preferences for punctuation, etc.) so that the phrases recognized by the system are musical ones.

2) Systematic research and creative exploration is necessary in order to achieve the best fit between feedback and performance (for a given musical idiom) and to ensure that the musical objectives do not become secondary to achieving a "good looking graph".

REFERENCES

Annette, J. *Feedback and Human Behaviour*. Penguin Books, 1969.

Arblaster, A. "The Evaluation of a Programming Support Environment." *Psychology of Computer Use*, Academic Press, London, 1983.

Austin, K. "The Midi Connection, Part II." MacTutor, 1984, 1 (12).

Beatty, A. "Bill Evans: Portrait of his Life as a Jazz Pianist and Musical Analyses of his Style". *Proceedings of NAJE Research*. 1986.

Bengtsson, I., and Gabrielson, A. "Analysis and Synthesis of Musical Rhythm." *J. Sunberg (Ed) Studies of Music Performance. Publications issued by the Royal Swedish Academy of Music*, 1983, Vol. 39.

Burnsed, V., Price, H. "Improvisation and Evaluation." *Proceedings of NAJE Research*, 1984.

Coker, J. *Improvising Jazz*. Englewood Cliffs: Prentice-Hall, 1962.

Dannenburg, R., and Mont-Reynaud, B. "Following an Improvisation in real Time." *Proceedings of the 1987 International Computer Music Conference*, Urbana: CMA, 1987.

Deutsch, D. *Grouping Mechanisms in Music*. New York: Academic Press, 1982.

Fraisse, P. *Rhythm and Tempo*. New York: Academic Press, (1982).

Free, J. Towards an Extensible Data Structure For the Representaion of Music on Computers. *Proceedings of the1987 International Computer Music Conference*, Urbana: CMA, 1987.

Gabrielson, A. "Interplay Between Analysis and Synthesis in Studies of Music Performance and Music Experience." *Music Perception* , 1985, 3 (1).

Gates, A., Bradshaw, J. "Effects of Auditory Feedback on a Musical Performance Task." *Perception and Psychophysics*, 1974, 16(1).

Giuffre, J. *Jazz Phrasing and Interpretation*. New York: Associated Music Publishers, 1969.

Good, M., Thomas, S., Whiteside, J., George, P. "User-Derived Impact Analysis as a Tool for Usability Engineering." *CHI'86 Proceedings*, Toronto: 1986.

Heller, J. "Electronic Graphs of Musical Performance: A Pilot Study in Perception and Learning." *Journal of Research in Music Education*, 1969, Vol. 17.

Henderson, A. *Improvisation in the Recorded Works of Coleman Hawkins (1932-34)*. Toronto: Masters Thesis, York University, 1981.

Henderson, M.T. "Rhythmic Organization in Artistic Piano Performance." *Studies in the Psychology of Music Vol. IV*. Iowa City: University Press, 1936.

Henderson, M.T., Tiffin, J., Seashore, C. "The Iowa Piano Camera and its Use." *Objective analysis of musical performance, University of Iowa Studies in the Psychology of Music Vol. IV*. Iowa City: University Press, 1936.

Hortin, J., and Teague F. Use of Visualization in Problem Solving: A Comparison of American and African Students. International Journal of Instructional Media, Vol. 11(2),1983-84.

Howell P., Powell, D. "Delayed Auditory Feedback with Delayed Sounds Varying in Duration". *Perception and Psychophysics* , 1987, 42 (2).

Lamb, M. "Andre Tchaikovsky meets the Computer: A Concert Pianist's Impromptu Encounter with A Musicianship Teaching Aid." *Int. J. Man-Machine Studies* , 1978, Vol. 10.

Lincoln, H. Uses of the Computer in Music Composition and Research. *Advances in Computers*, 1972, Vol. 12 .

Longuet-Higgins, H.C. "Perception of Melodies". *Nature*, 1976, Oct., Vol. 263.

Loy, G., Abbot, C. Programming Languages for Computer Music Synthesis, Performance and Composition. *Computing Surveys*, 1985 17(2).

MIDI Manufacturers Association. *MIDI 1.0 DETAILED SPECIFICATION*, Los Angeles: International MIDI Association, 1988.

Moore, R. "The Dysfunctions of MIDI." *ICMC Proceedings*, 1987.

Ostransky, L. *Understanding Jazz*. EngleWood Cliffs: Prentice Hall, 1977.

Papich, G., Rainbow, E. "Research in the Performance Practices of Musicians." *Psychology of Music*, 1975, 3 (1).

Patterson, B. "Musical Dynamics." *Scientific American* , 1974, 231(5).

Pennycook, B. "Computer-Music Interfaces: A Survey." *Computing Surveys*, 1985, 17 (2).

Politzer, R. *Teaching French: An Introduction to Applied Linguistics*. New York: John Wiley & Sons, 1965.

Povel, D. "Temporal Structure of Performed Music: Some Preliminary Observations."*Acta Psychologica* 1977, Vol. 41.

Pressing, J. "The Micro- and Macrostructural Design of Improvised Music." *Music Perception*, 1987, 5(2).

Reeves, S. "Techniques of Teaching Jazz Improvisation." *Proceedings of NAJE Research.*, 1986.

Rinzler, P. "Techniques of Advanced Improvisation -- Creating Syntax." *Proceedings of NAJE Research*, 1982, Vol. 2.
Rizzo, P. *Creative Melodic Techniques Used in Jazz Improvisation*. Cleveland: Phil Rizzo, 1973, Vol. 1.
Rose, R. "Computer Assisted Swing." *Jazz Educators Journal*, 1985, 17 (3).
Seashore, C. *Psychology of Music*. New York: McGraw-Hill, 1938.
Sherman, H. "Instrumental Jazz Techniques: Jazz Phrasing and Articulation." *NAJE Educator*, 1977, 9 (4).
Simon, H., and Sumner, R. "Pattern in Music." *Formal Representation of Human Judgment*. Kleinmuntz, B. (Ed). New York: John Wiley & Sons, 1965.
Sloboda, J.A. "Music Performance." *The Psychology of Music*, New York: Academic Press, 1982.
Small, A. "An Objective Analysis of Artistic Violin Performance." *Objective analysis of musical performance, University of Iowa Studies in the Psychology of Music Vol. IV*. Iowa City: University Press, 1936.
Stein, E. *Form and Performance*. London: Faber and Faber, 1962.
Sundberg, J., Frydén, L., & Askenfelt, A. "What tells you the player is musical? An analysis-by-synthesis study of music performance." *J. Sunberg (Ed) Studies of Music Performance. Publications issued by the Royal Swedish Academy of Music*, 1983, Vol. 39.
Tucker, J. Regimens of Informative Feedback. *Improving Human Performance Quarterly*, 1979, 8 (3).
Vercoe, B. "The Synthetic Performer in the context of Live Performance." *Proceedings of the International Computer Music Conference*, San Francisco: CMA, 1984.
Wessel D., Bristow D., Settel Z. "Control of Phrasing and Articulation in Synthesis." *Proceedings of the International Computer Music Conference*, San Francisco: CMA, 1987.
Williams, K. "A Method for the Computer-Aided Analysis of Jazz Melody in the Small Dimensions." *Annual Review of Jazz Studies*, 1985, Vol. 3.
Wills, B., Mackenzie, C.L., Harrison, D.B., Topper, T.N., Walker, G.A. "On the Measurement of Pianists' Keyboard Performance". *9th International Congress of Biomechanics*, 1983.
Zumbrunn, K.F. "Charlie Parker: Stylistic Features of His Blues." *NAJE Educator*, 1977, 9 (4).

THE REPERTORY GRID AS A VEHICLE FOR EXPLORING TEACHER/PUPIL PERSPECTIVES ON LOGO

W. Bell, E.S. Harri-Augstein
Centre for the Study of Human Learning
Brunel University
Uxbridge, UK

The Logo programming language provides an artificial intelligence based learning environment. Papert terms this environment a microworld, whereby children learn through play; they explore how they themselves think. Logo enthusiasts claim it to be a revolutionary educational tool influencing cognitive development. They argue that it is analogous with the thinking process; a similarity existing between the building up of Logo programs and that of thought structures. Evidence to justify these claims is lacking. Verification is difficult as standardised testing procedures are not compatible with the technique employed when using Logo. Most available evidence is anecdotal.

The purpose of this research is not evaluation, but to see how pupils and teachers perceive and reflect on the use of Logo in the classroom. It is concerned with the process of learning, not the product. The major tool employed is the conversational use of the repertory grid; although other methods such as observation, questionnaire and formal secondary sources will also be utilised. Grids are being elicitated from pupils and teachers in England and eventually from similar respondents in Canada. The grids will be analysed using the Integrated Repertory Grid Suite of Programs, developed at the Centre for the Study of Human Learning at Brunel University, England.

UTILISATION DE LA GRILLE DE KELLY À TITRE DE MÉCANISME D'ÉTUDE DES POSSIBILITÉS ENSEIGNANT/ÉLÈVE SUR LOGO

Le langage de programmation Logo permet de constituer un milieu d'apprentissage basé sur l'intelligence artificielle. Papert donne à ce milieu le nom de micromonde, dans lequel les enfants apprennent par le jeu et explorent leur propre façon de penser. Les partisans de logo soutiennent que ce langage constitue un auxiliaire didactique révolutionnaire influant sur le développement cognitif; ils y voient une analogie avec le processus de réflexion et établissent un rapport entre la création de programmes Logo et la structure de la pensée. Rien ne permet cependant d'étayer vraiment cette hypothèse. La vérification présente des difficultés puisque les procédures normalisées d'essai ne sont pas compatibles avec la technique d'utilisation de Logo. Les preuves à cet effet sont dans une large mesure circonstancielles.

La présente recherche n'a pas pour objet de procéder à une évaluation mais de voir comment les enseignants percoivent et comprennent l'utilisation de Logo dans les salle de classe. Elle vise le processus d'apprentissage et non le produit lui-même. Le principal outil est l'utilisation interactive de la grille de Kelly; on peut également faire appel à d'autres méthodes comme l'observation, les questionnaires, et l'utilisation de sources secondaires officielles. Des grilles sont aussi sollicitées d'élèves et d'enseignants en Angleterre et, éventuellement, d'autres répondants de même type au Canada. Elles seront analysées au moyen du Integrated Repertory Grid Suite of Programs, élaboré au Centre for the Study of Human Learning de l'Université Brunel, en Angleterre.

THE REPERTORY GRID AS A VEHICLE FOR EXPLORING TEACHER/PUPIL PERSPECTIVES ON LOGO.

Wynne Bell and E.Sheila Harri-Augstein.

Centre for the Study of Human Learning,
Brunel University, Uxbridge, Middlesex, England.

ABSTRACT

The Logo programming language provides an artificial intelligence based learning environment. Papert terms this environment a microworld, whereby children learn through play; they explore how they themselves think.

Logo enthusiasts claim it to be a revolutionary educational tool influencing cognitive development. They argue that it is analagous with the thinking process; a similarity existing between the building up of Logo programs and that of thought structures. Evidence to justify these claims is lacking. Verification is difficult as standardised testing procedures are not compatible with the technique employed when using Logo. Most available evidence is anecdotal.

The purpose of this research is not evaluation, but to see how pupils and teachers perceive and reflect on the use of Logo in the classroom. It is concerned with the process of learning, not the product. The major tool employed is the conversational use of the repertory grid; although other methods such as observation, questionnaire and formal secondary sources will also be utilised.

Grids are being elicited from pupils and teachers in England and eventually from similar respondents in Canada. The grids will be analysed using the Integrated Repertory Grid Suite of Programs, developed at the Centre for the Study of Human Learning at Brunel University, England.

APPLICATION OF ARTIFICIAL INTELLIGENCE.

The influence of an artificial intelligence (AI) style to the use of computers in education would ensure its usage in a more intelligent manner. While the proponents of AI research to education are concerned with propagating this influx, it is being approached in two ways. At one end of the continuum there are those who are researching intelligent tutoring systems (ITS), and at the other end, those interested in Logo style environments (Self 1986). While discovery learning may be an improvement on didactic teaching, many pupils require guided discovery learning. Sleeman (1985) considers that even with the aid of 'microworlds' (the most notable of the computational ones being 'turtle' graphics) most pupils learning Logo require a great deal of teacher support. He suggests that an ITS could be written to provide the required supportive environment. Hence the attempt to combine exposition and exploration.

Yazdani(1986) also advocates a 'happy medium in the spectrum, whereby systems are built incorporating good features of both approaches'. Sleeman (1985) illustrates the connection between expert systems and ITS by stating that ITS 'attempt to model the expert's (the teacher's) knowledge of a subject-domain and pedagogical techniques.' Perhaps teachers and pupils themselves could converse systematically and, taking on the role of knowledge engineer, produce a form of expert system incorporating their knowledge of Logo. This could be very simple initially, because what the children need to know to get started with Logo is minimal, but then it could be extended as their knowledge needs of Logo grow. Reynolds, (1988) considers it beneficial to teach some expert systems in the classroom, albeit the use of a microCODIL package is advocated.

SELF ORGANIZED LEARNING AND THE LOGO ENVIRONMENT.

The research described in this paper features an investigation into the Logo type environment. The proponents of the Logo philosophy are considered by a number of those in the field of ITS research to be the revolutionary faction of AI. They follow Piaget in observing learners as builders of their own intellectual structures, adopting the 'learning without being taught' paradigm. They lay many claims to the use of Logo in schools.

A study of the philosophy of Logo reveals a strong relationship to the Self-Organized Learning (SOL) methodology employed at the Centre for Human Learning (CSHL), Brunel University.(Thomas, Harri-Augstein 1985) Neither hold the behaviourist's view that learning can be measured exclusively in terms of performance. Learning, being a concept, is not directly observable. Only a change in personal understanding, or the structure of experience, linked with performance, is considered to be sound evidence of learning. The behaviourists have been criticised for the lack of meaningful learning employed. The SOL and Logo proponents consider that one learns from experience together with the personal meaning that is attributed to it. To obtain personal meaning involves a process of reflecting upon the experience. Reflective learning then raises awareness of the process of learning. This is clearly portrayed with the use of the 'turtle', a programmed robot, used in Logo. As a tool it is appropriate and personal to children. When they wish the 'turtle' to move in a specified way, they know how to carry out the required moves themselves; they reflect on this and then they teach the 'turtle' to carry them out.

With SOL and the Logo methodology, the emphasis is on learning as a process, not a product. This is in contrast with the nomothetic approach which focuses on test performance, and is akin to the drill and practice type of educational software.

SOL stresses the importance of a learning conversation to engender the growth of learning. Conversation abounds amongst children while they are programming in Logo. Papert (1980) recalls animated conversation centering on the children's personal knowledge. He states, 'we see a kind of epistomological reflection accompanying the activity'.

PERSONAL CONSTRUCT THEORY.

In this theory Kelly (1955) provided a model in order to facilitate understanding of the manner in which people construe, i.e. place an interpretation on, events. He presented his theory in terms of one fundamental postulate and twelve corollaries. Pope and Keen (Pope, Keen 1981) indicate how certain corollaries can relate to the various educational ideologies. Kelly perceived learning as a personal exploration and that in order to participate, the teacher must gain some sense of what is being seen through the child's eyes (Kelly 1970).

THE REPERTORY GRID.

The repertory grid was developed as a practical means of utilising the theory embodied in personal construct theory. This is the tool that is used to facilitate SOL, combined with its conversational use. The process has parallels with a structured interview, but it is formalised by the use of a two dimensional array to contain relationships of elements - events pertaining to the subject matter, and constructs - ways of thinking about these events. The elements must be representative of the person's experience of the area being investigated. Meaning is attributed to the elements by specifying relationships between them, their similarities and differences, hence the resulting bipolar constructs. The remaining elements are then assigned to the poles of the constructs.

CONVERSATIONAL ELICITATION OF GRIDS.

To understand the views of another individual, the conversational approach is necessary. Thomas and Harri-Augstein have developed the learning conversation as a method for teacher and learner to jointly reflect upon, review and develop the quality of learning. The grid is a tool that can be recruited to amplify this conversation. While interviewer bias can be a problem when eliciting grids conversationally, as with the structured interview, the grid reduces this bias through its standardised procedure; it is formally structured. The conversation is based around the respondent's thoughts and feelings on the grid topic. The elements and constructs are elicited from the respondent in their own terms. During and after the elicitation, the teacher/researcher encourages reflection, review and development of understanding, again in the respondent's terms. As the conversation develops, the researcher/teacher may offer his/her own understanding of the topic as a contribution, but it is the respondent's view that is of primary importance.

POPULATION.

With this research being on-going, the ultimate population may well differ from that initially envisaged. To-date the research subjects have been teachers and their pupils, mainly in the age range 7 to 13. Older physically handicapped pupils have also taken part. Initially the schools were state schools in the vicinity of Bedford, England. For critical case analysis, an independent school, and an adjustment unit have also featured in the research.

During the Summer of 1988, education advisors in Nova Scotia kindly offered much help, to enable subsequent research to take place in schools in the Province. This should enable a comparative study.

METHODOLOGY.

Educational advisors were contacted to determine suitable schools to contact, and questionnaires were sent. Preliminary telephone conversations, then preluded the visit to selected schools. The decision was to provide shared sets of elements. This was accomplished by involving teachers and pupils in a brief discussion to enable the identification of these elements. The teachers and children were to complete the grids using the conversational method. For the children the elements were individually written on cards. The cards were then presented in random arrangements of triads to each pupil to elicit their constructs. The ensuing conversation was taped on each occasion.

It is envisaged that a similar procedure will be followed in Canada.

PEGASUS SOFTWARE.

PEGASUS is the acronym for Program Elicits a Grid And Sorts Using Similarities. This program provides an alternative method to elicit the grids. It allows triads to be user chosen, rather than randomly selected. Besides conversationally eliciting the grid, this software offers interactive feedback and analyses the results. It was not viable to use this software in the selected schools due to the existing type of hardware.

GRID TOPICS.

At the outset the purpose for completing the grid has to be determined. Four grids were to be used initially and the main aim of each follows.

1.To place Logo in perspective with other educational software.
2. To obtain perceptions on the various claims that have been attributed to the use of Logo.

3. To reveal thoughts on the various facilities offered by the Logo language.
4. To elicit ideas on the various approaches to introducing Logo in the classroom.

Elements were subsequently selected to allow the specified purpose to be achieved.

DIFFICULTIES ENCOUNTERED.

1. Despite the presumed popularity of Logo in the classroom, much time had to be devoted to finding schools that were using it substantially; in some cases it was being used at a very superficial level. Also, there was little usage by pupils in the 13 to 18 age range.
2. The conversational method for eliciting repertory grids from the teachers was particularly difficult to carry out. It was apparent that these teachers did not have the time available. They could allow selected pupils to leave the class to complete grids, but naturally could not follow this procedure themselves. As a temporary measure, grid forms were produced, together with instructional literature, to enable grid completion without the need for assistance to be at hand. This distance approach has so far not produced a satisfactory response rate.

GRID ELICITATION.

All respondents were requested to complete grids, containing common elements, to aid subsequent comparison using relevant computer software, as specified under grid analysis.

A quickly emerging fact showed that the apprehension originally expected by attempting to elicit grids from young children was misfounded. But, it was soon evident how time consuming the grid procedure was; hence, a large number of respondents was not feasible. It proved beneficial with pupils to make three or four hour-length visits, if possible, for each person. This allowed them sufficient time to fully explain their constructs and prevent concentration lapse. The difficulty lay in condensing their rather verbose explanations to brief statements for construct pole descriptions, whilst ensuring that they were in agreeance with the result.

The pupils have proved to be better respondents at completing grids than the teachers. This points to the benefits arising from the use of the conversational technique. The pupils mainly looked on the procedure as a form of game. The teachers stated that they consider the repertory grid very difficult and time consuming to complete; hence the majority of the grids to-date have been obtained from pupils.

GRID ANALYSIS SOFTWARE.

Initially each grid will be analysed using the FOCUS computer software in order to show the pattern of personal meaning contained within it. Grids obtained at a later date will be analysed using the CHANGE computer software to reveal how the person's view of the topic varies over time. Grids will eventually be compared using the SOCIOGRID computer software.

SCHOOL VISITS.

These visits revealed that the method employed by the teachers in using Logo in the classroom differed from school to school. Teacher intervention was nil at one school; the children using Logo thought up problems between themselves and were mainly left to find any errors without teacher help. A reverse situation was evident at yet another school. The teacher always told the children what to do and provided help in correcting errors. With the handicapped pupils at the special school, the procedure involved much teacher interaction, as may be expected. In all schools visited, the use of Logo was confined to the graphics facility.

PRELIMINARY FINDINGS.

1. TEACHER COMPLETED QUESTIONNAIRES.
Total number sent : 53
Number returned : 34

(i) LOWER SCHOOLS-3 out of 24 schools contacted used Logo, (age range 5-9 years).
(ii) MIDDLE SCHOOLS-7 out of 14 schools contacted used Logo, (age range 9-13 years).
(iii) UPPER SCHOOLS-8 out of 15 schools contacted used Logo, (age range 13-18 years).

COMMENTS RECORDED RELATING TO LOGO:
Interested in learning about it:time factor.
Aids development from concrete to abstract.
Examination pressures:syllabus targets.
Cost, 40-50 machines required.
Assume it is a programming language.
Nobody seems to know what it is for.
Very unlikely that its use will become widespread.
Needs to be first taught in middle schools.
Applicable to middle schools-slow learners.
A limit to how long pupils remain motivated.
Pupils enjoy Logo, they take BASIC more seriously.
Expensive:difficult-except for graphics.

2. SCHOOL VISITS.
(i) LOWER AND MIDDLE SCHOOLS.
The children from the lower schools showed a lack of confidence, appreciating teacher direction in the use of Logo. Where the children had spent more time in pre-planning, i.e drawing pictures and writing down program instructions prior to using the microcomputer, they were more enthusiastic about using Logo, and completing grids.

All considered they had to think more about problems using Logo and while they thought it was easy at first, it soon got more difficult. Logo was liked because besides using it to help with mathematics, they were allowed to draw fun things. While they said they got frustrated by getting things wrong in other topics, they were not bothered by getting it wrong using Logo; considering it a challenge trying to get it correct.

(ii) SPECIAL SCHOOL.

The conversational mode was very important for these handicapped pupils; they were often inclined to offer constructs without too much thought behind their selection.

(iii) ADJUSTMENT UNIT.
It had been suggested that it may be impossible to expect these pupils to complete grids. This was far from being so; the pupils appreciated the one-to-one contact and gave very intelligent responses. Much usage of the computer in this unit enabled the pupils to have a greater knowledge of the various types of educational software than in other establishments visited.

(iv) INDEPENDENT SCHOOL.
A more serious attitude to the use of Logo prevailed; it was mainly considered from a mathematical viewpoint. The pupils appreciated its use as a change from the usual teaching method employed.

3. TEACHER COMPLETED GRIDS.
To-date only one grid has been completed from the 15 issued.

4. PUPIL COMPLETED RAW GRIDS.
(i) LOGO IN PERSPECTIVE WITH OTHER ED. SOFTWARE.
Constructs revealed that the pupils considered Logo a different approach-they being in control, and not being told, that it involved much thinking and planning and was more fun.
(ii) INTRODUCING LOGO IN THE CLASSROOM.
Some pupils liked deciding what to do themselves, but others thought it easier and saved time when the teacher told them what to do. All liked to work with friends; this was not always allowed with other school work. They mainly liked to work without the teacher's help, if possible. Pre-planning was considered time-consuming by some, but others thought they were more likely to get the program correct quicker by this method.

5.GRID ANALYSIS.
Grids are presently in the process of being elicited, and as yet, have virtually not been computer processed.

CONCLUSION.
Relating to the questionnaires; the survey was small and was carried out in February 1987, in the Chiltern area of England. At that time it would appear that there was surprisingly little usage of Logo in those schools which responded. The present difficulty experienced in finding schools which were making substantial use of Logo suggests that the situation has not altered. Requesting teachers to complete questionnaires at a future date may endorse this.
Initial impression was that more information was being obtained from the conversation arising during the elicitation of the grid than from the resulting grid itself. But, experience from previous research, which was dependent on methods other than grid usage, showed that extracting detail by conversation alone, compared unfavourably with that employing the combination of conversation and grid technique. Facts emerged without the problem of having to carefully phrase questions in order to obtain the required information. The children in particular were at ease and engrossed when thinking about the grid, and seemed to find it quite easy explaining their feelings about their experiences. Often they were controlling the conversation, and what was stated was in their own terms.
To gain an insight, using the grid technique, into how teachers perceive Logo has not been possible, as yet. The same does not apply with the pupils. The responses arising from the elicitations leads to the interim conclusions that they perceive Logo as something that makes them think more, that they consider fun, that enables more discussion and communication with other children, and that they are more in control using it than with other school work.

BIBLIOGRAPHY

Kelly,G.A.(1955). The Psychology of Personal Constructs, Vol.1,2. W.W.Norton &Co.Inc, New York.

Kelly,G.A.(1970). Behaviour as an Experiment. In Perspectives in Personal Construct Theory. (D.Bannister, ed.) Academic Press, London & New York.

Papert,S.(1980). Mindstorms: Children,Computers and Powerful Ideas. New York: Basic Books.

Pope,M.L.& Keen,T.R.(1981). Personal Construct Psychology and Education. Academic Press. London & New York.

Reynolds,C.F.(1988). Introducing Expert Systems tp Pupils. In Journal of Computer Assisted Learning, Vol.4, No.2. Blackwell Scientific Publications.June 1988.

Self,J.A.(1986). Artificial Intelligence-its Potential in Education and Training. Fifth Canadian Symposium on Instructional Technology, Ottawa, April 1986.

Sleeman,D.(1985). AI and Education: Two ideological positions. In The Quarterly Newsletter of the Society for the Study of AI and Simulation of Behaviour No. 55-56 Autumn-Winter'85.

Thomas,L.F. & Harri-Augstein,S. (1985). Self-Organised Learning. Foundations of a Conversational Science for Psychology. Routledge & Kegan Paul plc.London.1985.

Yasdani,M.(1986). Artificial Intelligence and Education: A Critical Overview. In Artificial Intelligence and Society. (K.S.Gill, ed.) J.Willey & Sons Ltd.

C.A.R.E.S. (CONCEPT D'APPRENTISSAGE DE LA ROBOTIQUE DANS L'ENSEIGNEMENT DES SCIENCES)

R. Pelchat
Les robots pédagogiques C.A.R.E.S. Inc.
Montréal, PQ

C.A.R.E.S. (Concept d'Apprentissage de la Robotique dans l'Enseignement des Sciences) est un nouveau concept d'apprentissage qui propose à des élèves de 9 à 12 ans de CONCEVOIR, d'ASSEMBLER, de TESTER et de PROGRAMMER des robots dans un cadre de résolution de problèmes. Au niveau secondaire, le logiciel et le contrôleur de C.A.R.E.S. sont utilisables en physique, en initiation aux sciences informatiques et en initiation à la technologie. La démarche d'apprentissage s'articule autour de 4 objectifs: 1) la conception, 2) l'assemblage et la mise à l'essai, 3) l'automatisation ou la délégation à l'ordinateur, et 4) l'exécution et l'évaluation. Au cours élémentaire, de la 4e à la 6e année, tous les élèves de la classe participent à la résolution d'un problème technologique à l'aide des robots qu'ils ont construits à cette intention. Chaque équipe de 6 élèves apporte sa solution originale au problème proposé. C.A.R.E.S. propose pour chaque degré une situation d'apprentissage distincte:

DEGRÉ	NIVEAU D'EXPERTISE	SITUATION D'APPRENTISSAGE	GUIDES	
			maître	élève
4e	Débutant	Le Robot enfouisseur de déchets nucléaires	oui	oui
5e	Intermédiaire	Le Robot jardinier	oui	oui
6e	Avancé	Le Robot analogique	oui	oui

C.A.R.E.S. (COMPUTERIZED APPROACH TO ROBOTIC IN ELEMENTARY SCHOOL)

C.A.R.E.S. (Computerized Approach to Robotics in Elementary School) is a new learning concept which allows students between the ages of 9 and 12 to CONCEIVE, ASSEMBLE, TEST and PROGRAM robots in a problem solving environment. At the high school level, the software and control unit of C.A.R.E.S. are used in physics, introduction to computer sciences and initiation to technology. The learning process revolves around 4 main topics: 1) the conception, 2) the assembling and testing, 3) the automation or transfer of control to the computer, and 4) the execution and evaluation. In grade school, 4th to 6th grade, all students participate in solving a technical problem with the help of robots built by them for this purpose. Teams of six students present their own solution to a given problem. C.A.R.E.S. proposes different learning environments according to various age groups (or grade):

GRADE	LEVEL OF DIFFICULTY	LEARNING ENVIRONMENT	MANUALS	
			teacher	student
4th	Beginner	The nuclear waste processing robot	✓	✓
5th	Intermediate	The gardener robot	✓	✓
6th	Advanced	The analogical robot	in preparation	

C.A.R.E.S. (Concept d'Apprentissage de la Robotique dans l'Enseignement des Sciences)

Roger Pelchat

Directeur de l'école St-Damase, (C.É.C.M.)
Auteur de C.A.R.E.S.

C.A.R.E.S. est un concept d'apprentissage favorisant l'introduction de la robotique scolaire dans l'enseignement des sciences avec des élèves de 9 à 12 ans.

Il privilégie l'atteinte de la moitié des objectifs terminaux prescrits par le programme officiel et propose trois niveaux d'expertise échelonnés sur le deuxième cycle du cours primaire (4e, 5e, 6e années). Chaque niveau d'expertise est documenté par un guide pédagogique et un guide d'activités.

- Le Robot enfouisseur, niveau débutant.
- Le Robot jardinier, niveau intermédiaire
- Le Robot analogique, niveau avancé.

Le niveau avancé (6e) compte douze situations d'apprentissage différentes, chacune documentée par un guide d'activités:

Le Bras articulé
L'Aquarium automatisé
L'Incubateur automatisé
Les Tours de Hanoï
Les Feux de circulation programmés
Le Robot mobile autonome
La Maison sécuritaire
L'Électro-aimant
La Chaine de production automatisée
Le Tapis roulant
La Serre automatisée

INTRODUCTION

Plusieurs préoccupations sont à l'origine de notre concept.

Premièrement il veut favoriser le DÉVELOPPEMENT de la PENSÉE SCIENTIFIQUE. Tout au long de son cheminement l'élève formule des hypothèses, les vérifie par l'expérimentation et tire les conclusions qu'il juge pertinentes. Dans ce cadre pédagogique, axé généreusement sur la résolution de problème, l'enfant peut échafauder SA solution. C.A.R.E.S. contribue donc, selon nous, au développement de la pensée scientifique par l'apprentissage et l'utilisation systématique de la démarche expérimentale.

Deuxièmement parce qu'il propose à l'élève du primaire de CONCEVOIR, d'ASSEMBLER, de TESTER et de PROGRAMMER des mécanismes agissant dans un ENVIRONNEMENT structuré, il veut favoriser chez lui une meilleure compréhension de l'environnement technologique dans lequel il évolue. La plupart des innovations contemporaines font appel à l'informatique. La compréhension théorique et surtout pratique de concepts, tels que ceux de «programme» et de «logiciel», s'avère donc importante pour l'élève.

Une bonne façon de faire réaliser à l'élève l'implacable logique d'un programme, c'est de lui en faire rédiger un. Que ce programme serve en même temps d'algorithme

° ensuite, à **travailler COMME les objets** (reproduction du réel connu)

° enfin, à **travailler SUR les objets** (construction d'unereprésentation symbolique)

Par OBJET, nous entendons ici tout ce qui peut être intégré aux mécanismes conçus par les élèves et qui contribue à la résolution du problème proposé.

Tous les enfants suivent un CHEMINEMENT semblable. Cependant, il importe de souligner deux variables importantes :

1° Le TEMPS DE PASSAGE d'un seuil à l'autre varie d'un individu à l'autre.
2° Tous les élèves n'atteignent pas le troisième seuil d'apprentissage.

Avec C.A.R.E.S., nous pouvons observer:

- Que la robotique constitue une formidable interface cognitive favorisant le passage de ces trois seuils d'apprentissage.
- Que parfois, la prise de contrôle de l'équipe est assumée par l'élève qui franchit le troisième seuil (leadership fonctionnel).
- Que l'élève garde toujours son intérêt car la robotique scolaire fait beaucoup appel au SAVOIR-FAIRE: manipuler, dessiner, construire.
- Que l'attrait de la culture technologique est généralement accru chez la plupart des élèves.
- Que les filles développent une nouvelle attitude devant les questions technologiques.Construire et manipuler ne seront plus à priori des activités réservées aux garçons.

DISCUSSION

La Boucle ouverte

En robotique pédagogique, le contrôle de procédés physiques est réalisé en BOUCLE OUVERTE. Celle-ci se déroule comme suit:

1- Un CONTROLEUR (ordinateur) émet une commande, généralement sous forme numérique, au mécanisme du robot: un bras articulé ou un outil quelconque.

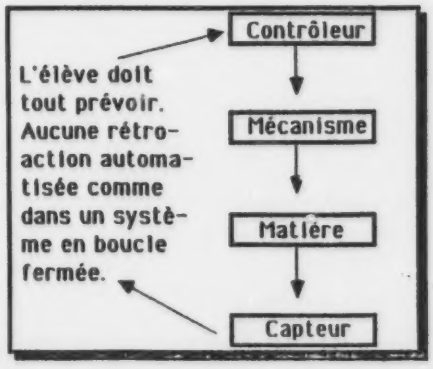
2- Un MÉCANISME obéit à la commande du contrôleur et agit sur la matière à transformer.

3- Cette modification de la matière est détectée et mesurée par des CAPTEURS, à condition que l'élève l'ait prévu dans son programme. Ceux-ci informent le contrôleur des effets mesurés: température, forme, humidité, contact, luminosité, champ magnétique, etc.

4- L'élève doit inclure dans son programme la COMMANDE suivante en fonction de la réaction PRÉVUE du capteur.

5- Le contrôleur, programmé par l'élève, envoie une nouvelle commande au mécanisme(rétroaction).

Schématiquement, on pourrait représenter la BOUCLE OUVERTE comme suit:



de contrôle pour des robots qu'il a conçus, assemblés et testés, voilà que tout un univers technologique s'ouvre à lui.

Troisièmement C.A.R.E.S. veut proposer un concept d'apprentissage où l'ordinateur sera le processeur d'un mode de contrôle déterminé par l'élève. Cette préoccupation de l'ordinateur «outil d'apprentissage» a toujours influencé le développement de C.A.R.E.S. et constitue un des facteurs importants à son origine.

Notre concept innove en offrant à l'élève un support informatique (logiciel et contrôleur) adapté et facile d'utilisation. Ce support technique vise à lui donner le plein contrôle de son processus d'apprentissage.

Quatrièmement il veut, à travers la robotique scolaire, privilégier une approche où **l'heuristique** et **l'algorithme** se complètent dans le but de résoudre un problème technologique. Dans la pratique, les élèves utilisent une spirale séquentielle de raisonnements hypothéticodéductifs et d'heuristiques pour solutionner une suite plus ou moins sophistiquée de problèmes, problèmes qu'ils ont eux-mêmes inscrits dans l'architecture de leurs mécanismes puisqu'ils en sont les concepteurs.

L'intérêt de C.A.R.E.S. réside essentiellement dans une démarche d'apprentissage où l'élève **conçoit, assemble, teste** et **programme** son robot. C'est une démarche globale dans laquelle il résout un problème mais demeure tout au long, maître des opérations puisqu'il est l'auteur de sa solution. À lui de corriger, à l'intérieur d'une boucle ACTION/RÉACTION constamment renouvelée, les différentes faiblesses techniques et logiques contenues dans l'idée originale de son robot.

Une distinction s'impose cependant quant au degré de performance visé dans le cadre de ce concept. Nous sommes dans un champ particulier de la robotique: la ROBOTIQUE PÉDAGOGIQUE. Cette dernière se distingue de la ROBOTIQUE INDUSTRIELLE en ce qu'elle met l'accent sur la quantité et la qualité des apprentissages plutôt que sur la performance des machines développées. En CONCEVANT, ASSEMBLANT, TESTANT et PROGRAMMANT ses robots, l'élève franchit divers seuils cognitifs qui lui permettent de se "construire" une représentation symbolique de la situation d'apprentissage qu'il exploite et par analogie, d'améliorer sa compréhension du réel.

L'enseignant, quant à lui, trouvera une formidable plate-forme pour l'observation et l'évaluation des apprentissages: heuristiques, qualité et originalité des solutions recherchées, trouvaille astucieuse ou finesse d'exécution d'un programme. La finalité de la robotique pédagogique s'oppose donc à celle de la robotique industrielle: le SUJET préoccupe plus le pédagogue que l'objet. La machine n'intéressera l'enseignant que dans la mesure où elle le renseignera sur le cheminement cognitif de l'élève.

Ajoutons enfin que C.A.R.E.S. est implanté dans plusieurs écoles du Québec et permet, en conséquence, d'offrir aux collègues un concept éprouvé, un logiciel de programmation simple et efficace, un contrôleur logique/analogique très performant, des guides pédagogiques et d'activités pour chaque situation d'apprentissage.

RÉSULTATS

Dans l'élaboration de SA solution au problème proposé, l'élève franchit TROIS SEUILS D'APPRENTISSAGE successifs.

Il apprend :

* en premier, à **travailler AVEC les objets** (familiarisation)

Dans un système d'asservissement en BOUCLE OUVERTE, l'élève doit prévoir toutes les ACTIONS de son robot, ce dernier ne pouvant rétroagir de façon autonome.

La BOUCLE OUVERTE constitue une situation d'apprentissage très riche car elle exige que l'élève définisse toute la partie "intelligente" de son MÉCANISME. Toute ACTION d'un MÉCANISME sur l'ENVIRONNEMENT, toute RÉACTION du CAPTEUR et toute RÉTROACTION, doivent être prévues dans son programme. Aucun automatisme intelligent, autonome et rétroactif. L'élève a la responsabilité intégrale de ses décisions.

Tout le matériel de C.A.R.E.S. est distribué au Québec par la firme:

Les Robots pédagogiques C.A.R.E.S. Inc.
10450, Meunier
Montréal, Qc
Canada, H3L 2Z4
(514) 385-3925

RÉFÉRENCES

BRULÉ, Michel, *Micro-informatique appliquée*, Module 6, les micro-systèmes et la robotique, Québec, 273 pages.

COIFFET, Philippe, *la Robotique, principes et applications*, Hermès, Paris, 1986, 433 pages.

EN COLLABORATION, *la Recherche en intelligence artificielle*, Coll. La Recherche, Ed. du Seuil, Paris, 1987, 377 pages.

EN COLLABORATION, *La Robotique* Time-Life, Amsterdam, 1987, 128 pages.

Journal officiel de la République, *l'Utilisation de la robotique dans la productivité et ses perspectives d'avenir*, Paris, 1982 pp. 381-447

LEMOIGNE, Jean-Louis (ouvrage coordonné par) *Intelligence des mécanismes, mécanismes de l'intelligence*, Fayard, Paris, 1986, 368 pages.

LOPEZ, Pierre et FOULC, Jean-Numa, *Introduction à la robotique* Editests, Paris, 1984, 318 pages.

PROJECT SPECIAL LINK: TELECOMMUNICATIONS LINKING HOMES, HOSPITALS, SCHOOLS AND LIBRARIES

R. Mason
Ontario Ministry of Education
Willowdale, ON

J. Cameron
North York Board of Education
Willowdale, ON

This paper will discuss a telecommunications project designed to link hospitalised and homebound students with their local schools in North York, Ontario, Canada. Due to a wide range of medical conditions the students in the project are unable to regularly attend their local schools. The project uses computer technology, FAX equipment and telecommunications software to allow students, parents and teachers to participate in an electronically humanised community. Through such strategies as group based problem solving and project participation, students aged from twelve to twenty-one enrich each other's academic, social and emotional growth. Other areas discussed include staff and curriculum development. The paper also focuses on the often complex family dynamics and the change process that emerges as this type of educational intervention attempts to decrease the isolation of program participants. These complex dynamics emerge as the perception of the student's role changes within the family from one characterised by social isolation and total dependence, to that of increasing involvement through active participation in broader more independent interpersonal contacts.

PROJET SPÈCIAL LINK: LES TÉLÉCOMMUNICATIONS RELIENT LES MAISONS, LES HÔPITAUX, LES ÉCOLES ET LES BIBLIOTHÈQUES

La présente communication décrit un projet de télécommunications qui permet de relier les étudiants hospitalisés et les étudiants immobilisés à la maison avec leurs écoles et qui a cours à North York (Ontario), au Canada. En raison de divers problèmes de santé, les étudiants qui participent au projet sont incapables d'assister régulièrement aux cours dispensés par leurs écoles. Grâce à l'informatique, au télex et à un logiciel de télécommunications, ces étudiants, leurs parents et leurs professeurs participent à une collectivité humanisée par l'électronique. Par le biais de diverses stratégies, telles la résolution de problèmes par groupe et la participation à des projets, des étudiants de 12 à 21 ans encouragent mutuellement leur croissance scolaire, sociale et émotive. L'étude porte également sur le personnel d'intervention et sur l'élaboration des programmes scolaires nécessaires. Elle place aussi l'accent sur la complexe dynamique familiale et sur les processus de changement que met en lumière ce type d'intervention pédagogique destiné à réduire l'isolement des participants. La complexité de la dynamique familiale découle du fait que l'étudiant, qu'on percevait comme étant socialement isolé et en situation de totale dépendance, s'affranchit de plus en plus activement grâce à un réseau de liaisons interpersonnelles.

PROJECT SPECIAL LINK

TELECOMMUNICATIONS LINKING HOMES, HOSPITALS, SCHOOLS AND LIBRARIES

Ron Mason: Educational Officer, Central Region, Ontario Ministry Of Education, Central Ontario Region, Suite 3201, 2025 Sheppard Ave East, Willowdale, Ontario Canada M2J 1W4

John Cameron: Project Manager, North York Board Of Education, 5050 Yonge St., Willowdale, Ont., Canada, M2N 5N8, 416-225-4661, FAX 416-223-0930

As a result of advances in medicine, increased political awareness and legislation requiring equal opportunity for all students, students are now entering school systems with needs that many school systems, until recently, generally assumed they could not integrate into their programs. The notion of who has the right to a full and well rounded education and the ability of technology to provide delivery mechanisms for educational programs, has forced many educational policy makers and planners to reconsider their positions. This is particularly the case with those groups of students who who cannot attend school on a regular basis.

Consequently, school Boards are faced with the challenge of providing equal opportunity access to educational services for such children. This is particularly challenging since in many cases these students cannot attend school in a traditional manner and for medical reasons are confined to their hospitals or homes. However, the power and flexibility inherent in telecommunications software and hardware equipment can allow these and other students to overcome many of the physical limitations imposed upon them by distance and medical conditions.

During 1987 The North York Board of Education in Ontario, Canada, through Ron Mason the Co-ordinator of Special Education, anticipated this need and responded by proposing a pilot project entitled Project Special LINK designed to link home-bound students with a school in their community. The students are considered as members of specific classes within their schools. The telecommunications component of Project Special LINK allows scheduled interaction with teachers and classmates thus providing educational and social enrichment.

An organizational framework for the policy, organization and operation of the pilot project was subsequently defined. The first stages of project implementation began during January 1988. The diagram at the end of this paper illustrates the interrelationship of project components.

Project Hardware and Software Organization

Users have two main telecommunication options. The first option involves a direct link from school to wherever the student is residing. For example, if the student is at home teachers and students can operate in the chat and mode and discuss the lesson content or issues that are currently taking place within the classroom. Chat mode allows real time discussion and interaction where questions can be asked and any content can be explained in detail as required. This mode can continue until the student at home has enough information to continue working on an independent basis. When this condition is met teachers and students in the classroom can then send files that can contain assignments or more information on the topic under discussion.

The student at home need only leave the computer they are using in an answer mode and the files from the school are automatically transferred. Files can be sent at anytime to the home computer as long as the student at home remembers to leave it in the answer mode. The communications software being used also allows whoever sent the data from the school to catalogue the home computers directory to ensure that no errors have taken place in the transmission.

The second telecommunications option involves sending data via a third computer located in the school systems main library. The project has four dedicated telephone lines that allow users flexible access to the system. This central node is a Unisys ICON central fileserver with a dedicated memory of 70 M.B. It is on twenty-four hours each day seven days a week. The Iconet software program available from Dr Manfred Hanke at Nipissing University North Bay Ontario, that runs this system has six powerful components. In order to use the system users are assigned passwords and E-Mail boxes. The entire Iconet program has six components:

Autoanswer: This program causes Iconet to switch communications parameters to match those of any incoming call from different types of computers and software.

Mail: Users on the system can send E-Mail messages to each other. Each time a person logs on the system checks for mail and automatically lists any that is waiting to be read. An interesting feature contained in the system is the a beep will go off if someone is trying to reach you if you are on the system talking to a second person. Iconet also informs each user when the mail they sent has been read.

Conference: Conferences can be open or closed, and may be held in real or delayed time.

Chat: This option allows for real time communication. It appears to foster small group interaction and expression. Many users who are otherwise intimidated with the telephone or face-to-face encounters enjoy the chat mode

Learn: This component allows subject specialists to develop interactive CAI programs that can be user for distance learning.

Utility: This option basic file manipulation where files can be sent to and received from other locations.

Project Operation And Administrative Overview

Many large school boards have traditionally handled the needs of home-bound students at home or in hospital by sending an itinerant teacher five hours a week. Our goal here has been to go beyond this and re-define the relationship the home-bound student has with the school. Instead of accepting a representation that defines the student as physically and psychologically removed from the school, we have rejected this in favour of a model that believes student-school separation as a result of distance is irrelevant Having done this we enhance the students self perception as a necessary daily participant within an identified grouping within a school. Where possible we have tried to provide leadership opportunities for those students working at a distance so that they gain extra status as a member of a school class.m It is also vital that the community be involved so that realistic educational approaches that would have transfer potential to future workplace settings. Further, a clearly defined need must be identified to justify the resource allocation necessary to staff and run projects like Special LINK. The evolution of this project for the first ten months of its operation was done on a volunteer basis. As a result of this it was determined that this form of telecommunication application was in fact cost-effective.

Other insights that were gained were that it is critical to have partners from other areas of education such as psychological services. This form of support we now believe to be critical since many of the families with seriously ill children are often in crisis. In fact it is not an exaggeration to say that despite our reliance on technology it is **PEOPLE THAT MAKE THINGS HAPPEN through their creative application of technology**. Corporate sponsors were also included from the start. Eventually the project's potential was recognized and since Jan 88 it has began to grow.

Project operation takes place within policy and procedure, curricular and staff development frameworks. Two committees act as policy and procedure formulating bodies with a highly diversified composition representing political, curricular, business, administrative, parental, teaching and medical interests. These committees have a very practical orientation. For example, one of the business representatives from the Unisys corporation helped facilitate the hardware acquisition process. The committees are also organized to provide support in other areas such as curricular and staff development.

Operational management of the project is the responsibility of the project manager who serves on both the committees mentioned. Another important role within the project is that of case manager. Within each school a teacher has the role of a case manager for the student(s) at home or in hospital. The case manager organizes a team working with the parents, students and the project manager to define a communications schedule and curricular objectives designed in accordance with students available time. A critical component in this phase of the project is direct parental co-operation and participation in the working team. Once an agreement to establish a link has been arrived at and curricular objectives defined, the project manager designs an individualized training program to meet the often unique needs of each home/hospital school link. The training cycle, including hardware acquisition, on average, lasts between two and three weeks.

Project Operation And Administration Overview.....*continued*

The progress of each location where the project is in operation is assessed on a regular basis to determine if the link established remains appropriate. It is projected that within the next five years that the number of potential candidates who will be able to make use of this type of educational service within the City of Toronto will be in excess of one hundred candidates.

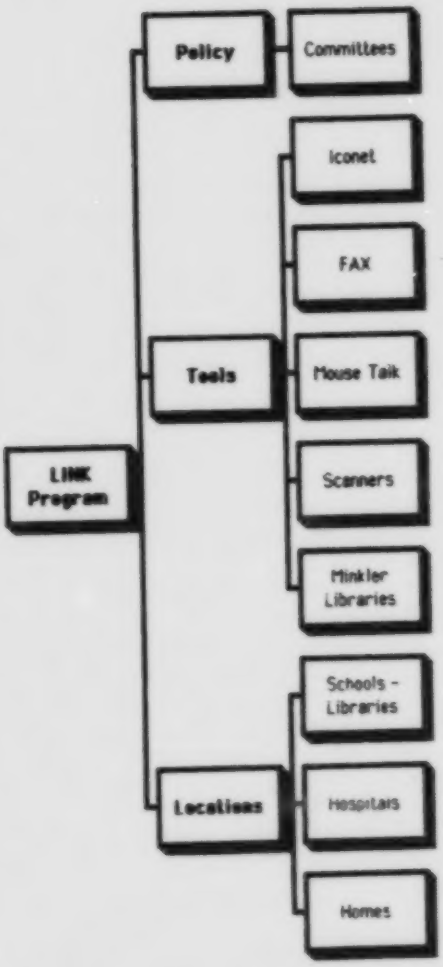
An important aspect of Project Special LINK is the connection with Special Education. In Ontario ,Special Education is mandated by legislation with clear regulations regarding identification, placement , programming and review of student progress. By identifying student participating in Special LINK it provides safeguards for schools system and student alike. Review is ensured and objective decisions regarding the selection of students to be involved area made by an independent identification committee.

Some Tentative Observations And Conclusions

It is now clear that societies assuming the title of enlightened cannot ignore the needs each type of handicap presents existing within such societies. The educational needs of students that are isolated from their peers in hospital, home or other locations must be considered within the framework of educational policy. A step that would enhance the quality of programs delivery for home-bound students that might be considered is the establishment of a closer dialogue between hardware and assistive devices manufacturers.

Secondly it must be clearly understood that meeting the needs of the home-bound student in general requires that the total family situation be taken into account from the very onset of educational planning and intervention. Families of seriously ill children have been under high levels of stress for often a period of years. The anxiety, fear and ambiguity of such situation can often require medical, psychological and sociological professional support. Educators must be aware of these needs and consult very closely with the families they work with and with any other professionals that might also be working with these families.

PROJECT LINK COMPONENTS



PRESCHOOL CHILDREN AS TEACHERS OF COMPUTER ANXIOUS ADULTS

L.A. Doran, M.F. Kalinowski
Dept. of Family Studies
University of New Hampshire
Durham, NH, USA

Results of a project in which children aged 3-5, enrolled in the UNH Child and Family Center, instructed college students rated as computer anxious, compared to traditionl training of college students delivered by adults in existing computer orientation sessions.

benefits to college students and to children will be explored and recommendations presented for making computer learning environments comfortable for both children and adults.

ENFANTS D'ÂGE PRÉSCOLAIRE QUI ENSEIGNENT À DES ADULTES DÉSIREUX D'APPRENDRE À UTILISER UN ORDINATEUR

Comparaison entre, d'une part, la formation donnée par des enfants de 3 à 5 ans inscrits au Child and Family Center de L'UNH, dans le cadre d'un projet, à des étudiants de collège désireux d'apprendre à utiliser un ordinateur et, d'autre part, la formation traditionnelle donnée par des adultes, dans le cadre de programmes existants d'initiation à l'ordinateur, à des étudiants de même niveau.

On examinera les avantages offerts aux étudiants de collège et aux enfants et on fera des recommandations pour rendre les milieux d'apprentissage par ordinateur plus agréables pour ces deux groupes.

PRESCHOOL CHILDREN AS TEACHERS OF COMPUTER ANXIOUS ADULTS

L.A. DORAN, M.F. KALINOWSKI

Department of Family Studies
University of New Hampshire
Durham, NH

INTRODUCTION

Often when adults learn to use computers for the first time, they may be anxious, perceiving the computer as a powerful machine, ready and waiting to frustrate them (Danzinger & Kraemer, 1986). It may be this same fear that deters large numbers of adults from learning the benefits of computers. On the other hand, many young children demonstrate considerable interest in computers, approaching them with comfort and confidence (Binder & Ledger, 1985). It has also been reported (Borgh & Dickson, 1986; Kull, 1986; Wright & Samaras, 1986) that young children tend to help and teach each other about computers, and that computers encourage development in young children (Davidson, 1989). With this in mind, the investigators tried to determine if it was possible for young children to introduce adults to computers and to reduce their anxiety in the process.

SUBJECTS

The experimental group included 14 preschool aged tutors and 17 college student trainees. Tutors were 48-60 month old preschoolers (8F,6M) from predominantly middle class white families, randomly selected from the University laboratory school. Trainees were undergraduate college students from a large, undergraduate general education course. From a pool of 82 college students responding to an initial questionnaire, 41 reported they experienced computer anxiety. Of this computer anxious group, 17 were randomly selected to participate in a one-on-one computer orientation session in which the preschoolers were the primary instructors. The control group included the remaining 24 college students who attended a computer orientation session offered through University Computing taught by an adult computer orientation instructor. All 17 students who worked with children rated themselves as being fearful/afraid of using a computer and as experiencing at least moderate anxiety when considering having to use a computer. Of the 24 students in the control group, 11 reported that they were fearful/afraid of having to use a computer and experienced at least moderate anxiety when considering using a computer, while 13 reported that they were not fearful/afraid, but they did experience at least moderate anxiety.

METHOD

After gaining parental consent, children participated in 30 minute computer training sessions in groups of 2-4 children once or twice per week over a period of six weeks. During this training, preschoolers were taught the parts of a Macintosh computer and how to use the program, Early Games (Springboard) for the Macintosh. Near the end of the training period, children were told that a group of adults who had never used computers before, or who were scared of using a computer, would like the children to help them learn to use computers. Children were each asked if they would like to teach an adult, and all responded that they would. Before teaching adult subjects, each child practiced teaching a peer how to use the computer.

In 17 one-on-one computer orientation sessions, preschoolers individually taught 17 computer anxious college students to use the Early Games software program. Each orientation session, lasting 25-30 minutes, took place in a private office in which the computer was positioned on a desk with chairs for child and adult. Each was videotaped and observed by the investigator, who remained in the room throughout the session, provided prompts or answered questions, and coded behaviors. Orientation sessions included labeling computer parts (i.e. mouse); demonstrating turning the computer on/off; inserting the disk; accessing and playing the games of the program; and removing the disk from the computer. At the end of each session, adults completed a survey about their experience which was designed to measure their degree of computer anxiety following the session and which provided feedback about their experience working with a preschool child. The preschoolers were also asked a series of questions by the primary researcher. Three weeks after each orientation session, a follow-up questionnaire was sent to each to the adults subjects to measure their degree of anxiety about the computer at that point and to reevaluate their feelings about working with a preschool child.

Control group subjects attended a 90 minute computer orientation session in a University computer cluster, conducted by an adult

instructor. The first hour included a lecture about parts of the computer, university rules and regulations, software, procedures for turning on/off computers and inserting disks, computer memory, and using printers. The final half hour was devoted to individual tutorial software programs for AT&T or Macintosh computers. Three weeks after the orientation session, each subject was sent a follow-up questionnaire designed to measure their degree of anxiety about using computers at that point and to get feedback about the benefits and drawbacks of that type of computer orientation program.

RESULTS

The majority of adults in the experimental group reported very little or no anxiety about using the computer within the session (94%), that the session was worthwhile (88%), that it made them feel less anxious about the computer (82%), and that they would feel less anxious the next time they used as computer (82%). Almost all (92%) of the responses to computer use questions were correct. Although none of the subjects reported that they would prefer a child as an instructor, only two (11%) reported, after the session, that they would have preferred an adult. The majority of children (94%) enjoyed the activity. None would decline to participate a second time. One child (6%) expressed some discomfort talking to an adult. If given a choice, about half of the children (47%) would prefer tutoring an adult rather than a peer. Adults provided considerable praise and attention to their tutors.

The majority of adults in the control group (57%) felt somewhat less anxious using a computer following the orientation session. While there was a range of opinion, in general the control group found their sessions too long, technical, boring, and not providing enough "hands on" experience. While this may have been due to their particular experience, one assumes this response is not atypical of many computer orientation sessions.

IMPLICATIONS

On the basis of this preliminary study, having computer anxious adults work with young children has several advantages. The adult is forced to become involved in the activity because young children may have problems reading the screen, adding or subtracting, or considering the implications of some responses. Observing a computer adept young child may help adults feel more confident because they see preschoolers able to accomplish the same tasks they will be required to undertake. Children generally interact with computers at a fairly slow pace, and this may encourage adults able to think more quickly and anticipate what may be coming next.

In generalizing the results to training by adults, one may consider potential advantages in creating a relaxed setting free of intimidation, where one starts with a relatively slow pace, uses short sessions in the initial introduction of computers, offers considerable hands-on experience with another peer, and begins with children's educational software as opposed to computer games or adult applications.

REFERENCES

1. Binder, S.L., & Ledger, B. Preschool Computer Research Project, Sheriden College. Ontario, Canada, 1985.

2. Borgh, K., & Dickenson, W.P. Two Preschoolers Sharing One Microcomputer: Creating Prosocial Behavior With Hardware and Software. Young Children and Microcomputers, P.F. Campbell & G.G. Fein (Eds.), Reston, VA, Reston Publishing, 1986.

3. Danziger, James N., & Kraemer, Kenneth L. People and Computers: The Impacts of Computing on End Users in Organizations, New York, Columbia University Press, 1986.

4. Davidson, Jane I. Children & Computers Together in the Early Childhood Classroom. Albany, N.Y., Delmar Publishers, Inc. 1989.

5. Kull, J.A. Learning and Logo, P.F. Campbell & G.G. Fein (Eds.), Young Children and Microcomputers. Reston, VA, Reston Publishing, 1986.

6. Wright, J.L., & Samaras, A.S. Playworlds and Microworlds, P.F. Campbell & G.G. Fein (Eds.), Young Children and Microcomputers, Reston, VA, Reston Publishing, 1986.

POLICY/POLITIQUE

STRATEGIES AND PRACTICES FOR INTERNATIONAL COLLABORATION OR EDUCATION SHOULD BE IMPORTED AND NOT EXPORTED.

W.R. Broderick, R. Fothergill
The CERES Trust
Brentwood, U.K.

Models for collaboration are urgently needed in the field of Information Technology (IT). All nations are rightly protective towards the characteristics of their educational practice that preserve their national and cultural distinctiveness. But few nations are themselves large enough - or even have a critical mass of expertise - to support a profitable national educational software industry. So collaboration is forced on us all.

The mechanics of collaboration require careful study. IT eases the communication problems, but how should we tackle the issues of cultural and curricular differences? How should we build up understanding of the differences between educational systems? At what stage should partnerships be formed to build strength into collaborative efforts?

Is the "not invented here syndrome" a fact of life - or a reaction of a teacher who needs to add some of his own approach and identity to the materials. Yet we see a flourishing commercial international software industry for business. We do not know where much of our PC software is written. Why are the experiences of the office and the classroom so different? Why are the differences between classrooms so great? What collaborative strategies are most effective in building bridges between dissimilar environments?

STRATÉGIES ET PRATIQUES DE LA COLLABORATION INTERNATIONALE OU LES SYSTÈMES D'ÉDUCATION DOIVENT ÊTRE IMPORTÉS PLUTÔT QU'EXPORTÉS

La technologie de l'information (IT) a un besoin pressant de modèles décrivant la coopération. Tous les pays protègent à juste titre les caractéristiques de leur tradition d'enseignement dans le but de préserver leurs particularités nationales et culturelles. Mais peu de ces pays sont suffisamment grands - ou bien on la masse critique d'experts nécessaire - pour supporter une industrie de rédaction de didacticiels rentable. Il faut alors recourir à la coopération.

La mécanique de la collaboration exige une étude attentive. La technologie de l'information aplanit les problèmes de communication, mais on doit se demander de quelle façon il faut aborder les différences culturelles et les divers programmes de cours. Comment s'y prendre pour modéliser notre compréhension des différences entre les systèmes d'éducation? À quelles étapes doivent être constitués les partonariats pour que se renforcent les efforts de collaboration?

Est-ce que le "syndrome cela n'existe pas ici" découle de la réalité ou est une réaction d'un professeur qui veut insuffler une partie de sa façon de faire et de ses préférences dans les outils de cours? Il est vrai que le marché international de la rédaction de logiciels de gestion est florissant, mais qui sait où nos logiciels PC sont rédigés. Pourquoi l'écart est-il si grand entre l'expérience des bureaux et celle des classes d'enseignement? Pourquoi un écart si important entre les classes? Quelles stratégies collaboratives conviendraient le mieux à réunir des milieux dissembables?

Strategies and Practices for International Collaboration
or
Education should be imported and not exported

W.R.Broderick & R.Fothergill
The CERES Trust, Brentwood, Essex, UK.

Educational Exchanges and collaboration.

Education, like language and culture, is widely held to be a central element of a societies identity. Indeed many would hold that education is inseparable from language and culture in that it is delivered by means of the former and in the context of the latter.

There has always been educational and cultural exchange on a limited scale. Aid programmes often feature educational exchanges, studentships and support programmes. These are not just altruistic gestures, but reflect the power of education as a tool for inculcating a set of values. Academic institutions, at the post school level, have always exchanged experts as a way of broadening their range of expertise.

At school level there has been relatively little exchange until recently. The age of cheap travel and electronic communications has however made new opportunities available for teachers and students to share their experiences. Exchanges of this nature will continue to grow in free societies.

Importing and exporting of educational materials is not uncommon. At the Higher Education Level it is very common with text books, the only limitations being language. Equipment is also quite widely exported. At the school level the tendency is for text books to be exported, but largely to nations with similar examination traditions and syllabuses.

But this free exchange is not universally welcomed. In some states only books and materials on an approved list are legally allowed to be used in schools. Indeed this is a reflection of how politicians fear the power of education.

Anxieties about the dilution of culture and the swamping of local dialect and languages by their more dominant and widely used neighbours are well known - and have been known to lead to civil disturbance in every continent.

Against this background it is understandably difficult to export educational software.

The new dimensions added by educational software.

With educational software the issues are brought in to sharp relief because the economics of production demand a larger market than that provided by one nation alone. Much of this software is targeted at the school, rather than post school level.

The situation is further complicated by the fact that in some states education is a local rather than national responsibility. To the outside observer at least, it would appear that some local juridictions find very strong reasons to be different from their immediate neighbours. Thus there are many more different "markets" than nations.

Educational software is likely to get more expensive as the education profession harnesses the advanced features of the newer manifestations of the technology and applies them to the learning process. This will only increase the need for educational materials to be shared more widely.

Language and culture are already a frequent barrier to such sharing. Differences in hardware standards and operating systems merely add a problem to an already difficult situation. Further problems arise different curriculum, different learning traditions, methods and styles of teaching.

Why bother?

A sensible approach to take when a difficult question is raised is "Does this problem need solving?"

There are several facets of this problem: economics, educational and technical.

If substantial investment in educational software is to be made, the larger burden to be carried by the smaller states makes collaboration essential for them for economic reasons. Recognition of the power of education is manifest in some of the more nationalistic and insular practices of governments but does not extend as far as generous funding.

The educational case for collaboration is strong. Mutual respect and understanding between nations and people can be developed through working

together in a spirit of genuine co-operation. Information technology and cheap and fast travel all tend to make the world a "Global Village" where we share many experiences through the media and through personal travel. Sharing some of these experiences in an unstructured way is helpful, but can give rise to frustration, because "others" behave in apparently strange ways. Working together in education to achieve common goals provides a focus for developing a professional and personal understanding and also a basis for broadening our understanding of how people learn.

The technical problems are just that, technical, and they will diminish in difficulty as the demand for moving software from one environment to another increases, just as they have with data formats in the past.

Where do we start?

People accept things in which they feel they have an interest but they reject things that they feel are imposed on them. "If you cannot export then you have to get the recipient to import".

Of course, it is impossible for everybody to contribute to everything. Careful planning and design can provided many opportunities for people to receive software that they feel that they would like to use, would like to tailor in some way to their needs or would like to share with colleagues during training sessions.

The first target therefore is to ensure the materials are acceptable to the teachers for their students. On the simple basis that local sensitivities are local issues, involvement in design by a broad spectrum of potential users is vital. Multinational teams soon learn of each others' approaches and styles if they have the task of building software that accommodates them. It is by this mechanism that acceptability is built into the initial thinking.

The second set of problems are easily handled when recognised. These are to do with the language differences. Accurate and sensitive translation is important. But designers have to leave enough screen space to accommodate languages that are more greedy for space than their own!

Then there are details of a cultural nature, the nuances of language culture and behaviour and also styles of signs, the side of the road on which people drive, the stars that can be seen in the sky at night.

For these reasons, consultation during the development process must be continuous. The programming team must be very sensitive to the things it can and cannot add "because they make sense and the designer forgot to mention them!"

Of course continuous interaction between the design group and the international consultants is very important. There is no room for the "I know best" attitude. Email makes consultation much easier and it is reasonably easy to Email programs.

Some classic mistakes have been observed. Sending documents round a circle for adding comments may be successful in a small local team working in a disciplined environment. It often breaks down when the team is spread across a continent, speaks 4 languages, consists of busy people and has members with very different cultural attitudes to time. Likewise, placing administrators, with other priorities between discussions and actions can lead to some very confused message, no messages at all - or a network of decision making that selectively leaves them out of the loop. This can also lead to unnecessary difficulties.

Finding the people

Thus a network of resourced, committed educationists, with a team of developers who are sensitive to cultural needs and who can communicate effectively via Email is required. Where are these people with the staying power to cope with the work and frustrations over a development cycle that may be 2 to 3 years long.

They will come from several different countries, work in different types of institution and specialise in different subjects. To select them it is necessary to draw a profile that can be communicated to a colleague overseas who speaks a different language and from a different culture. Developing the skill takes time and experience. Relying too heavily on the advice of some Ministries of Education leads to recommendations that are not necessarily appropriate.

Thus there is considerable value in an independent network of well connected people in different countries who can provide professional judgment and obtain good advice.

Resources

Maintaining this database of contacts, keeping up to date with many national developments and being able to act as a local node in this network is a time consuming task. It needs resourcing as an independent part of a national educational activity sensitive to both public and professional concerns.

There is little doubt that it costs more to write software on an international scale than on a local scale. But the benefits are considerable, both in educational and economic terms.

Anxieties about funding and managing international educational projects can now be confidently addressed as a result of increasing experience of successful collaborative work and the development of a project management methodology that minimises the risks.

The confident identification of compatible teams with the necessary expertise requires much more than looking in a database and asking a few friends. It needs a network of contacts in countries, each will need the time and resources to remain independent, knowledgeable about their national activities and resources and sensitive to the special contributions they nationally can make and obtain from collaboration. Such collaboration will bring many benefits to all parties, especially those who will reap the benefits of marketing materials to wider audiences or those who will be able to recruit more internationally sensitive staff to run their enterprises in the multinational and multicultural world of the 21st Century.

Such a network is slowly being created by the CERES Trust, an independent charitable international organisation. Of course this will cost money, some raised locally, some of internationally.

Finding the funding and resources for collaborative projects is already under way in order to develop the base on which co-operation between educators of all nationalities can be funded and developed in the future.

ONTARIO MINISTRY OF EDUCATION SPECIFIES ITS MICROCOMPUTER -- PART 3: MID-LIFE CRISIS

R.S. McLean
Dept. of Measurement, Evaluation and Computer Applications
The Ontario Institute for Studies in Education
Toronto, ON

Part I, in the 1983 Symposium, reported on the Ministry's Functional Requirements which specified a networked microcomputer system that it would support financially and for which it would develop curriculum-related materials.

Part 2, in 1986, described the "development years" during which the ICON computer appeared and the Ministry's software initiatives were under way.

Part 3 brings the story up to date. There are considerable accomplishments, including about 150 curriculum-related programs; innovative software which capitalizes on the network capabilities; and an infrastructure of over 30,000 student stations in more than 2,000 local area networks. But the Ministry has revised its specifications to allow more manufacturers to qualify, and is embarking on a major "Software Portability Environment" development to ease the movement of software between the various vendors' environments.

LE MINISTÈRE DE L'ÉDUCATION DE L'ONTARIO PRÉCISE LES CARACTÉRISTIQUES DE SON MICRO-ORDINATEUR -- PARTIE 3: LA CRISE DES 40 ANS

Dans la partie 1, lors du symposium de 1983, on a signalé les exigences fonctionnelles du Ministère; il était question notamment d'un micro-ordinateur connecté que le Ministère financerait et pour lequel il élaborerait des matériels reliés aux programmes d'études.

Dans la partie 2, en 1986, on a décrit les "années de développement" durant lesquelles l'ordinateur ICON est apparu et les logiciels du Ministère étaient en voie de production.

Dans la partie 3, on met à jour l'histoire. Il existe de nombreuses réalisations, y compris environ 150 programmes reliés à des programmes d'études, un logiciel novateur qui bénéficie des possibilités du réseau et une infrastructure de plus de 30 000 postes d'étudiants dans plus de 2 000 réseaux locaux. Cependant, le Ministère a modifié ses spécifications pour permettre à un plus grand nombre de fabricants de répondre aux normes établies et s'engage dans un programme de développement important en vue de faciliter l'échange des logiciels compte tenu des divers fournisseurs.

Ontario Ministry of Education Specifies its Microcomputer-- Part 3: Mid-Life Crisis

Robert S. McLean

Department of Measurement, Evaluation, and Computer Applications
Ontario Institute for Studies in Education
252 Bloor St. W., Toronto M5S 1V6

This is the third of a set of papers presented at these symposia which describe efforts by the Ontario Ministry of Education to develop an educational computing infrastructure within schools under its jurisdiction. The papers each present a triennial record of the Ministry's microcomputer initiatives, and attempt to articulate a set of principles which may underlie those initiatives. If such principles can be identified, they may help in understanding the Ontario initiatives and contribute to the selection of future courses of action, in Ontario and elsewhere.[1]

This paper derives its subtitle from the observation that the Ontario microcomputer and software development initiative is now well established -- perhaps into its "mid-life," between being an enthusiastic start-up and its ultimate retirement because computer technology has been applied appropriately in all aspects of education. To the end of 1988, about $170 million had been spent on the various aspects of this initiative (see Table 1), including about $125 million in provincial grants to school boards for equipment. Although it has always been subject to criticism, it is now far enough into its career to examine its accomplishments and perhaps wonder if its chosen career path is the right one. Indeed, several changes have been made and others proposed, but not yet realized.

The initiative must now deal with two major challenges: 1) providing curriculum-related software that will run on a wider variety of school microcomputers; and 2) ensuring effective use of that software.

TABLE 1 -- Expenditures on the Ontario Grant Eligible Microcomputer Systems (G.E.M.S.) Initiative
(In millions; data from Provincial Auditor, 1988)

Research, development, field testing (includes "lighthouse school" projects)	4.3
"REE" incentive grants to school boards to purchase conforming microcomputer systems	96.9
Educational software (includes Ambience user interface, Software Development Assistance Centre	24.9
Other direct expenditures (includes Computers in Education Centre)	4.9
Total in eight years to March 31, 1988	$ 131.0

(Expenditures in fiscal 1988: $48 million)

[1] These papers have been developed independently of the Ministry of Education. Interpretations expressed are those of the author and may not reflect the views of the Ministry. On the other hand, it appears that there is no other continuing attempt to chronicle the Ontario developments, so it is hoped that this series has some historical value as well.

Recording the History

There appears to be no official history of the Ontario Ministry of Education's microcomputer initiatives. This series of papers attempts to summarize them, and there are various official Ministry memoranda.

Another source of historical documentation is the *CALM Development* newsletter published monthly by the Software Development Assistance Centre (SDAC). SDAC was established early in 1985, at the Ontario Institute for Studies in Education, under contract to the Ministry of Education (Horn, 1986). Its mandate is to provide assistance to developers for the Ontario Grant Eligible Microcomputer Systems (G.E.M.S.): providing information about Ministry development programs, advising on pedagogical and technical design issues, helping solve technical problems, finding test sites, helping assure quality control of software, and disseminating information through its newsletters and public forums.

Four years of *CALM Development* newsletters have included a wide variety of articles representing developer concerns, Ministry policy and requirements, technical concerns, and announcements of product and service availability (see Table 2).[2]

Background

Part 1 of this series (McLean, 1983) reported on the origins of the Ministry's microcomputer initiatives. In March, 1983, the Ministry published its *Functional Requirements for Microcomputers for Educational*

Table 2: *CALM Development* Article Topics

44	Technical concerns	12	Instructional design
39	Forum announcements and reports	6	IBM EDNET
		6	Speech and sound
20	Development process	6	Lighthouse projects and field research
20	Ministry specifications		
13	Software evaluation	5	Ministry organization

[2] References to *CALM Development* articles will be indicated in parentheses as: (*CD*, month and year).

TABLE 3 -- Report Card on the "Ten Advances" cited at the 4th Canadian Symposium on Instructional Technology (McLean, 1983)

Advances cited in 1983	Author's current appraisal
Educational motivation: Each specification has an educational justification, tied to the Ministry's educational objectives.	Many of the specifications are still sound, and have permitted the Ministry to commission a level of software which otherwise would be unlikely; industry has caught up, however, with machines of comparable sophistication.
Ergonomic design: Suitability for educational environment found in schools, including safety concerns, robustness and adjustment.	Safety and durability specifications are useful criteria for selecting possible equipment. However, the ergonomics specifications have not had any great effect, or if they have, the manufacturers have not explained their interpretations well.
Hardware support for user interface: bit-mapped display, pointing device and speech synthesis.	Bit-mapped display and pointing device have been crucial contributors to Ministry software. Originally, speech synthesis was difficult and of poor quality; few programs have used it.
Software support for user interface: Icon-based interaction, windowing.	Has made a qualitative difference in software, but has not been adopted consistently; no uniform "look and feel" developed.
Operating system: a "UNIX-like" (this term has since been dropped) operating system with concurrent multi-tasking and modularity.	Has given great flexibility to software and a learning challenge to developers' programmers. Hierarchical file system has increased perceived complexity of system. Over-use of modularity has sometimes degraded performance, however.
Network environment: fileserver for economy and convience and inter-task communication for new organization of educational activities.	Concepts of shared resources and connectivity are becoming understood and useful, but network management tasks have been daunting for many. Multi-node messaging has opened new possibilities for multi-person educational activities.
Telidon T-500 presentation standard: a try at addressing the problem of incompatible graphics standards across computers.	This is the least successful of the specifications and has been dropped. No satisfactory implementation was provided and little software developed.
Portable implementation of software: Implementation in C, and standard toolkit.	These were promising starts, but since the toolkit was not ported to other G.E.M.S.s, portability remains just an objective.
State of the art processor: the 16-bit processor requirement in 1983.	The G.E.M.S. still have a reasonable level of processing power today.
Open-ended, directed further development: anticipation of stage II Ministry specifications (originally expected in 1984).	Stage II was proposed in 1987; eventually a course based on evolution prevailed. Stage IA specifications still require that vendors accommodate further Ministry developments such as the SPE.

Use in Ontario Schools -- Stage I (OMOE, 1983). That document described the hardware and software which would be required in microcomputers for which the Ministry would develop educational software and which it would subsidize with incentive grants to school boards which purchased them.

The 1983 paper suggested that the Ministry's specifications were derived from its principles as well as pressure from the field, and listed ten "advances" over the typical educational microcomputers then extant. The ten are summarized in Table 3, along with an assessment of them six years later.

As summarized in Table 4, the first Ontario Approved Educational Microcomputer (OAEM) was the *ICON*, developed by CEMCORP and now marketed by Unisys. An IBM model, the EDNET, was qualified in March, 1986. The term "OAEM" was discontinued and "G.E.M.S." (for Grant Eligible Microcomputer System) introduced.

Educational software consistent with Ontario curricula has always been the driving force behind the initiative, even though the public's focus has been on the

TABLE 4 -- Summary of events recorded in previous CSIT papers. (Mclean, 1983, 1986)

1979-1980	Microcomputers increasing; enthusiasts urge Ministry to join "computer revolution"
Fall, 1981	Ministry study of "ideal educational computer" results in functional specifications; consortium of Canadian manufacturers (CEMCORP) commissioned to produce it.
March, 1983	More specific Functional Requirements released by Ministry; single company (CEMCORP) given purchase guarantees to produce computer.
Fall, 1983	CEMCORP produces prototype ICON, test phase begins.
First paper (McLean, 1983) given at 4th CSIT, Winnipeg.	
Spring, 1984	ICON accepted as Ontario Approved Educational Microcomputer (OAEM)
January, 1985	First ICON software released by OESS.
Second paper (McLean, 1986) given at 5th CSIT, Ottawa.	

hardware. Beginning in 1984, the Ministry issued calls for proposals for development of exemplary software and provided an elaborate set of requirements, reviews, and support (Smye, 1989), which has resulted in nearly 200 items now released or in development, virtually all for the ICON (Table 5). Many of these items are quite substantial, some comprising several megabytes of program and data. Additionally, the Ministry built a user interface, called the Ambience©, which provides a multi-user, cooperative environment in which most Ministry software runs. A number of papers and guides available from SDAC provide details of the Ministry's conventions and requirements in software development and field testing.

There have been related Ministry activities in computers in education; they can only receive brief mention here. Several "lighthouse" schools were designated, each working looking at a particular aspect of computer use (*CD*, April 1987, Jan. 1988), following an extensive previous study (Carmichael, *et al.*, 1985).

Two experiments linking many schools through computer communications were carried out under the name EDAN (*CD*, June 1988); the computer conferencing aspect is being continued by a project called CHIMO (Wilton, 1989). The Ministry also funded research into a local area network based class-wide database system which is built on the principles of "intentional learning", called CSILE (Scardamalia, *et al.*, 1989). In 1988, TVOntario surveyed 1280 Ontario teachers, finding little awareness of OESS software and a belief that the ICON was not the most suitable computer for educational use (Gates, 1988).

Software Portability Environment

One of the key concerns of the Ontario microcomputer initiative has always been that educational software used in schools should be useable regardless of the model of microcomputer found in a particular school.

TABLE 5 -- Software Release Schedule

	English	French	Other
January 1985	8	4	Ambience
May	3	3	Converted public domain
October	12	13	Ambience 1.2
January 1986			Public Domain
April	16	5	
October			Word processing and spreadsheets
November	9	10	
December			Ambience 1.4 field test
February 1987	1	0	
April	4	1	
June	3	2	
September	27	3	Ambience 1.4
March 1988	9	0	
June	4	2	
October	20	4	

This idea prompted the first set of *Functional Requirments*, which sought a solution by standardizing equipment across schools.

But since the standardization was in terms of function rather than specific hardware and software, it quickly became evident that many different and incompatible microcomputers could meet the *Functional Requirements*, but would not accept the same programs. This was illustrated when the IBM EDNET qualified as a G.E.M.S. The EDNET and the ICON computers meet the Ministry's *Functional Requirements*, but they differ in memory map, display resolution and operating system; to this day most Ministry programs do not operate on the EDNET. Those that do were converted at significant cost.

The Ministry proposed to overcome this incompatibility by means of a "Common Environment" which would span the two machines; it would be a subset of a comprehensive "Software Portability Environment" (SPE) in the next stage of the Ministry's *Functional Requirements*, which were already in preparation. A preview was given at a developers' meeting, June 4, 1986, where it was reported that work was in progress on specifying the Common Environment (*CD*, June 1986). That work never was made public.

The Stage II document was published as a draft for comment on April 1, 1987, and entered a period of discussion until September, 1987. During that time there seems to have been considerable agreement that some sort of software portability environment was a reasonable approach to the fragmented market for educational software in Ontario and elsewhere. Overall, however, the Stage II document fared badly because it was seen as requiring too grand a microcomputer system. Stage II had retained most of the conceptual components of Stage I, extending them to a more generalized network capability and relying more heavily on industry standards which are now developing.

The SPE was to be the cornerstone of Stage II. It would provide a *layered* software architecture as shown in Table 6. The basic idea is that all applications software is implemented using only the characteristics of the upper layers; no reference is permitted to the layers below the "Educational Applications Software Interface" (EASI) layer. The EASI layer is

TABLE 6 -- Layers of the Software Portability Environment (From OMOE, 1987)

7 The Ambience; Custom user interfaces
6 Lessonware and non-tool applications
5 Languages; Operating system emulators
4 Tools: GKS; Text editing tools; database tools; Sound tools

--- Educational Applications Software Interface (EASI) ---
BOUNDARY

3 'C' compiler; Low-level EASI primitives and utilities
2 Machine-specific interface layer
1 System hardware

implemented specifically for each vendor's microcomputer, using machine-specific techniques. But EASI presents a consistent set of software capabilities to the higher levels. Higher level software is written in the ANSI standard C language and makes no reference to any machine specific characteristics. Thus, porting of applications software to a new machine's EASI environment is simply a matter of recompiling the program using the target machine's C compiler.

The SPE provides a software environment specified by the Ministry of Education, and implemented by the equipment vendor. The vendor may implement the lower levels in any combination of hardware and software that is appropriate, but the functionality of the EASI must be exactly as specified by the Ministry. The EASI consists of the lower three layers of figure 6; these include the system hardware (layer 1), the interface functions (e.g., an operating system or BIOS), and the interface layer (layer 3) which contains the 'C' compiler, device drivers and low-level EASI primitives and utilities. Level 3 is the lowest level that an applications program can access, and such access is only in terms of the standardized EASI specifications. These functions are common to all conforming microcomputer systems.

The higher levels will be consistent over all vendors' equipment, since they will be implemented using the EASI conventions. Layer 4 includes building blocks of general applicability across many applications. Level 5 contains higher level tools, which would be accessible on their own or from higher level programs. Level 6 contains the bulk of the educational applications software, and the top level constitutes user interface "shells" which represent the unifying structure seen by the user (such as the Ministry's "Ambience" user interface).

The EASI layer provides ten categories of functions which can be called from higher levels. The function definitions are based on the proposed IEEE 855 Microprocessor Operating System Interface Standard (Jackson & Cowan, 1984) with additional functions appropriate to the additional functionality found in the G.E.M.S. (OMOE, 1987, p. 7.8). The ten functional categories are shown in Table 7.

TABLE 7 -- Classes of functions in the EASI, with examples (after OMOE, 1987, p.7.8 ff)

Memory management: dynamic allocation and freeing of memory during execution.
Time management and scheduling: timing within programs and scheduling program events.
Data transfer: file input and output, including access conventions and privileges for the network fileserver resources.
Process management: starting and stopping tasks and obtaining information about tasks.
Process synchronization and communication: message passing between tasks, locally and over a wide area network.
System inquiry and control: determining the configuration of the host system and manipulating it.
Exception handling: handling error conditions and user requests to terminate execution of a program.
System accounting: login and logout and use of resources.
Display graphics support: manipulation of the graphics resources and the various drawing primitives, including pixel-level support.
System error message suport: ability to provide alternative ways of handling error conditions which arise at the system level, and informing users about them.

New initiatives

Comments on stage II were received in September, 1987 (*CD*, Oct. 1987). After further consideration, including examination by a Minister's Council on Educational Technology, the Ministry abandoned it and issued *Stage 1A*. (OMOE, 1988, July; *CD*, Aug. 1988). Stage 1A reiterated most of the specifications of the original *Functional Requirements*, though there were notable changes (*CD*, Sept. 1988). One of those was the requirement that *1A* vendors subsequently implement the Software Portability Environment, with which the Ministry was going to proceed.

The Ministry solicited proposals to implement the SPE EASI layer from likely contractors in July, 1988. Although the work was to begin by mid October, as of February 28, 1989, the Ontario government had not yet awarded a contract.

The SPE EASI layer prototype was to have been implemented by summer 1989, and would be critical to any further Ministry of Education software development, if that software were to be usable across the multiple G.E.M.S. microcomputers which might be recognized as a result of *Stage 1A*. [3]

A further modification to the "Recognized Extraordinary Expenditure" (REE) grant policy was announced early in November, 1988. Boards of education would be allowed to spend up to 25% of their REE grant entitlement on equipment meeting another set of *Functional Requirements, -- Stage 1B* (OMOE, 1988, Nov.). These allow Stage 1A vendors to provide simpler stand-alone microcomputers which are appropriate for particular areas of education such as business or computer studies. In many cases these will be MS-DOS machines. Only stage 1A vendors qualify, to ensure that the service infrastructure is already in place across the province.

Additionally, school boards can now use up to 5% of the REE grant for *niche* technology -- e.g., CAD/CAM or applications for exceptional students.

A further 5% can be used for teacher training. This -- and a Computers in Education Centre reorganization -- seems to give more recognition to the need for specific steps in *implementation*. Until recently, learning about the software and technology had been seen as the teacher's professional responsibility, but now it is viewed as a responsibility also shared by the Minis-

[3] As this was being written, it was widely known that Commodore had qualified a system under *Stage 1A*, although the official announcement was awaiting a specific product designation. One or more additional vendors were believed to have submitted systems for qualification; Ministry policy is not to reveal any information until the vendor offers the specific product.

try and their software developers. It is somewhat ironic that a major study of the implementation process, completed in 1987 (Fullan, 1987, reviewed in *CD*, April 1988), still is not available to the public due to delays in providing a translation into French, as required by Ontario government policy.

The Auditor's Report

The Ontario Provincial Auditor chose Computers in Education as one of the major government programmes for review in 1988 (Provincial Auditor, 1988; *CD*, Jan. 1989). While some of the findings are open to criticism (Ragsdale, 1988), the report is an official "snapshot" and appraisal of the initiative.

From September, 1987, to March, 1988, the auditors visited 20 boards of education and 40 schools, interviewing principals, teachers, directors of education and superintendents of curriculum. They also visited the Computers in Education Centre, TVOntario, and SDAC. The report, released on November 30, 1988, included 11 pages (of 225) of findings, many negative; the Auditor's headings are listed in Table 8.

The Computers in Education Centre had already responded to some of these concerns, e.g., with *Stage 1B*. Some findings describe situations which, with hindsight, might not now occur, but which were part of the learning process associated with the new technology. Still other findings reflect a difference of opinion about the appropriateness of the Ministry's objectives and model. The Ministry's objective has been described in previous papers of this series: to provide advanced educational software, related to Ministry curriculum guidelines, covering a diverse sample of the curriculum and grade continua, encouraging "creative use" of computers, available across the province, and accessible to all teachers and students. It would seem that many of the auditor's findings and recommendations are in conflict with some aspects of those objectives, and are built on some other model.

Conclusion

The Ontario microcomputer initiative is at a crossroads, caught between the belief that its general objectives are proper and can be realized, and a number of conflicting realities. With nearly 200 substantial pieces of educational software licensed for use by all Ontario schools and about 30 000 G.E.M.S. which can run that software now in Ontario schools, there has been a significant effect. Among the probable effects are a larger group of participating teachers and a greater use of local area network-based common resource systems. Indeed, some of the controversy surrounding the initiative probably contributed to the general level of computer awareness among educators in the province (though they were not always happy with what they were learning). Much of the original vision has not been realized, but a major initiative of this scale -- derived from principles of education and visions of how to build the future -- merits recognition in the history of computers in education.

Table 8 – The Auditor's Findings*

Unique design added costs, delayed implementation.
Less costly computers would meet board needs.
Industry development objectives were not met.
Some good software was developed.
Some of the development was poorly planned.
Delays in approving software projects.
Ineffective field testing of software.
Insufficient emphasis on implementation.
Inadequate teacher training support.
More effectiveness research needed.

* Adapted from headings in (Provincial Auditor, 1988)

References

Carmichael, H.W., Burnett, J.D., Higginson, W.C., Moore, B.G., & Pollard, P.J. (1985) *Computer, Children and Classrooms: A Multisite Evaluation of the Creative Uses of Computers by Elementary School Children.* Toronto: Ontario Ministry of Education.

CD (various dates) *CALM Development.* Newsletter of the Software Development Assistance Centre, Ontario Institute for Studies in Education, 3rd floor, 252 Bloor St. W., Toronto M5S 1V6.

Fullan, M.G., Miles, M.B., & Anderson, S.E. *Strategies for Implementing Microcomputers in Schools: The Ontario Case.* Toronto: Ontario Ministry of Education, *in press* (to be available from the Ontario Government Bookstore, 800 Bay Street, Toronto M7A 1N8.)

Gates, D.B. (1988) *School survey on the use of microcomputers and the Ontario Educational Software Service (OESS).* Toronto: TVOntario, Evaluation and Research Report no. 6-1988-89. [ONTERIS ON05025]

Horn, E.J. (1986) Ontario's Software Development Assistance Centre addresses the needs of educational software development. *5th Canadian Symposium on Instructional Technology,* Ottawa. Pp.606-611. [*CD*, July, 1986]

Jackson, D.L. & Cowan J. (1984) The proposed IEEE 855 Microprocessor Operating System Interface Standard.*IEEE Micro.*

McLean, R.S. (1983) The Ontario Ministry of Education specifies its microcomputer. *4th Canadian Symposium on Instructional Technology,* Winnipeg, October. Pp.435-441 .

McLean, R.S. (1986) The Ontario Ministry of Education specifies its microcomputer. Part 2: The development years. *5th Canadian Symposium on Instructional Technology,* Ottawa, 601-605. [Reprinted in *CD*, June, 1986]

McLean, R.S. (1988) Local area networks in education: An infrastructure perspective. In Lewis, R. & Tagg, E.D. *Computers in Education.* Amsterdam: North-Holland. Pp. 259 - 264.

McLean, R.S. (1988-9, Dec.-Jan.) An educational infrastructure for microcomputers in Ontario. *T.H.E. Journal,* 79-82.

OMOE (1983, March) *Functional requirements for microcomputers for educational use in Ontario schools - Stage I.* Toronto: Ontario Ministry of Ed.

OMOE (1987) *Functional requirements for microcomputers for educational use in Ontario schools. Stage II (Draft only)* Toronto: Queen's Printer.

OMOE (1988, July) *Functional requirements for microcomputers for educational use in Ontario schools. Stage 1A.* Toronto: Ontario Ministry of Ed.

OMOE (1988, Nov.) *Functional requirements for microcomputers for educational use in Ontario schools. Stage 1B.* Toronto: Ontario Ministry of Ed.

Provincial Auditor. *1988 Annual Report.* Toronto: Queen's Printer.

Ragsdale, R.G. (1988) Reflections on the auditor's report. Unpublished ms.

Scardamalia, M., Bereiter, C., McLean, R.S., Swallow, J., & Woodruff, E. Computer supported intentional learning environments. *Journal of Educational Computing Research 5* (1), *in press.*

Smye, R. (1989, Dec.) The consequences of Ontario's Ministry of Education software environment. *ECOO Output, 9*(4), 29-32.

Wilton, J.A. (1989, Dec.) CHIMO - the new kid on the block. *ECOO Output, 9* (4), 23-25.

INSTRUCTIONAL TECHNOLOGY AND THE FEDS: LEADERSHIP ISSUES AND INITIATIVES

D.K. Griffin
Training Programs Branch
Public Service Commission of Canada
Ottawa, ON

A new team is being established within the Training Programs Branch of the Public Service Commission in Ottawa to help all federal departments make better use of new instructional technology for the training and development of public servants. Making effective use of computer technology for training requires a rethinking of existing definitions of communications technology, methods of supply and funding, and the relationship between learning and work. The introduction of new technology to the workplace has created new demands for training, as well as new training opportunities. Traditional trainers welcome the computer, as a powerful new tool, but many are fearful of the kinds of changes that it will bring.

This paper will describe, briefly, past and present situations and the plans that the Public Service Commission has for the future in this area. One of the first initiatives will be the creation of a demonstration or learning centre to evaluate and display new training products, including CBT and interactive video. The PSC intnds to be a focal point for all federal departments for information on services, products, and project management to help managers use new training technologies better.

LES TECHNIQUES D'ENSEIGNEMENT ET LE GOUVERNEMENT FÉDÉRAL: PRÉOCCUPATIONS ET INITIATIVES DE CHEF DE FILE

Une nouvelle équipe est présentement mise sur pied à la Direction générale des programmes de formation de la Commission de la fonction publique à Ottawa pour aider tous les ministères fédéraux à mieux utiliser la nouvelle technologie pédagogique dans la formation et le perfectionnement des fonctionnaires. Pour utiliser efficacement la technologie informatique dans les programmes de formation, il faut repenser les définitions de technologie des communications, de méthodes d'approvisionnement et de financement, ainsi que la relation entre l'apprentissage et le travail. L'arrivée de nouvelles technologies dans le milieu du travail a créé de nouvelles demandes de formation, de même que de nouvelles occasions de formation. Les animateurs de formation accueillent avec plaisir l'ordinateur comme outil nouveau et puissant de formation, mais bon nombre craignent les genres de changements que cet outil amènera.

Dans cet exposé, on décrira brièvement les situations passée et présente, ainsi que les plans d'avenir que la Commission de la fonction publique s'est donnés en la matière. L'une des premières initiatives sera de créer un centre de démonstration ou d'apprentissage pour évaluer et présenter les nouveaux produits de formation, y compris le matériel d'enseignement assisté par ordinateur et le matériel vidéo interactif. La CFP veut se faire, pour tous les ministères fédéraux, la première source d'information sur les services, les produits et la gestion de projets afin d'aider les gestionnaires à faire un meilleur usage des nouvelles technologies de formation.

INSTRUCTIONAL TECHNOLOGY AND THE FEDS: LEADERSHIP ISSUES AND INITIATIVES

D.K. Griffin, Ph.D.

Public Service Commission of Canada

Meeting New Needs

The Canadian Federal Government is facing significant new requirements for the training of public servants, at a time when all governments are looking for ways to reduce expenditures.

The most pressing of these is the need to train public servants in the use of PC software applications. In the current fiscal year we expect to spend $200 million on the purchase of these machines, compared with $25 million in 1984. The employees who will use these machines need to be trained. This will create not only an enormous increase in the need for training, but also an opportunity to use technology for training. Language training comes second in order of importance in terms of generic training needs of public servants. This includes both second-language skills, as well as improved first-language skills and the related benefits that follow, including improved communication and problem-solving skills.

Some General Observations

Federal departments have been experimenting with computer-based training technologies for several years. The major users have been National Defense and Revenue Canada. Based on my observation of the departments that have been experimenting with technology-based training over the past five years, I have made the following general observations.

1. Technology-based training is most easily introduced as a support for new computer systems, because the training need is clearly identified by the user, and the technology is available.

2. Off-the-shelf courseware should be used where available, because the creation of custom in-house software is expensive. The skills needed to create high-quality courseware are usually lacking.

3. Technology-based training usually gets better support from senior management and from end-users, than from systems people or trainers. The greatest opposition to new training technology in all environments comes from trainers or teachers.

I shall be describing later an example of how new training technologies have reduced training costs in a major US corporation. Given the critical importance of training for the economic well-being of all countries today, the federal government has an obligation to support the use, development, and improvement of training technologies that result in better, faster, and cheaper education and training. The government should be supporting new training technologies in both the public and private sectors, and should be actively promoting

the development of Canadian expertise that will make us more competitive in world trade. Given the major changes that are required to make use of new training technologies, this leads to a fourth major conclusion:

> 4. Using new training technologies requires a re-thinking of how training funds are allocated, and the tasks that trainers perform. To a major extent, training must become an integral part of working.

Current Federal Involvement

In order to help federal departments use technology-based training to meet new training needs, the Public Service Commission has recently set up a new Training Technology section within the Quality Assurance Program of the Training Programs Branch. The work of other federal departments is relevant to this project, and linkages are being formed. Industry, Science, and Technology supports industrial development and research in information technology; Employment and Immigration supports the retraining of workers; the National Research Council supports many related activities, including this conference; the Department of Communications supports both technological and cultural activities; National Defense trains military personnel, and Secretary of State, via the Jean Talon Project and the National Literacy Initiative, is supporting applications of learning technology to address cultural and social needs. The Public Service Commission, alone among these, has accepted the mandate to try to draw together the various related aspects of the technology, the training needs, the research and development initiatives, as well as the economic and social implications of the introduction of new training technologies.

Evaluation Role of PSC

The Training Technology Section of the PSC has set up a Training Technology centre in Ottawa, equipped with computers and laser-video technology, for the evaluation and demonstration of courseware which may assist departments in meeting their training needs. In the past, very little evaluation has been done of actual learner outcomes. This centre is currently evaluating the effectiveness of interactive video, CBT, and combinations of linear video and CBT. The centre will provide hands-on experience for departments exploring the use of technology-based training. It is our intention to help departments avoid the re-invention of wheels,(especially square wheels) and to facilitate the sharing of current information and expertise regarding hardware, software, evaluation, costs, as well as developing some general instructional design standards for courseware.

Training and Productivity

The current view in American business is that the competitiveness of industries for the next decade will depend more upon the skills and knowledge of employees than upon new technology or the availability of raw materials. (Japan is not rich in raw materials, for example.) This view does not come naturally to many traditional managers.

This view has been supported in two ways. First, analyses have been carried out of the relative success of U.S. companies in weathering the recession at the beginning of this decade. This has revealed that the companies which made a serious investment in improving the skills of existing employees fared far better than those which simply cut staff in response to reduced business. Secondly, some training managers in US companies have captured the attention of

managers by conducting surveys of employee skills within the various departments of their companies. When these survey results are presented to the section managers, this usually results in serious action plans to improve employee training.

Some studies appear to suggest that the general skill and knowledge level of the Canadian worker may be better than in the US. Whether this is true or not, we must not forget that the US school graduate and the US worker show up near the bottom on most international comparisons of skills, knowledge, competencies, and productivity. The US worker should no longer be our basis of comparison. We need to know about the productivity of workers in countries who represent our real competition, such as the countries of the Pacific rim. It would appear that in many instances, Canadians are less productive than some of these.

Role of Government

In the United States, the development of technology-based training was initially supported by funding by the federal government via the military. In the last two or three years, sources have funding have been found increasingly in the private sector. The Canadian military has also been a major user of technology-based training, but much of the training that they have purchased has come from the United States. Military data showing the extent of use of technology-based training have been much more available for analysis than data from large corporations, since many of these corporations have been unwilling to reveal information that could be useful to their competitors. Two important exceptions to this rule are Federal Express and IBM.

The IBM Example

The experience of IBM is instructive, because of the position of this corporation on the international business scene. IBM does such a good job at marketing its image that we all seem to be able to identify with this company in some way.

The Director of Training for IBM has just made public some interesting details of how IBM has used technology-based training, in a new book. I would like to repeat for you some of the salient facts that I think are relevant to other users of this technology. This may provide some relevant information for Canadian planners of training.

In 1979 IBM had decentralized their training function, in keeping with the then-current managerial philosophy. By 1984 IBM decided that they needed to get a handle on their training costs, and management of training was centralized again. IBM discovered that good information about the nature of training and its costs and benefits was not available through the company's management information systems. A study revealed that IBM was spending about $900 million per year on training; management decided that cuts were needed.

About $200 million was cut out of the training budget. At the same time, further analyses carried out by the company revealed that the sections of the company which had taken training seriously during the recession of the early 1980's had fared much better than those which did a poor job of training. Faced with the need to reduce the training budget and the recognition that training was essential in order to be competitive, IBM turned to technology-based training as a possible means of addressing both these needs.

Instructional Design Issues

Most users of technology-based training, including IBM, believe that the next significant stage in the development of technology-based training will be improvements in instructional design, rather than changes in technology. Good video-based and computer-based technology exists now that is being poorly used, in many cases. Most video, although labelled as "interactive", tends to be linear video interrupted by farily simple CBT drill. One of the major contributions that the Training Technology Centre in Ottawa can perform is to help establish in Canada some more interesting and more effective instructional design standards which will perform better than the approaches which currently prevail in the United States. These will include much better analysis of the learner's needs and situation. If the future success of technology-based training is to depend on the quality of instructional design, then Canadians can be competitive in the industry, since our instructional designers are already better than some others we could mention. And we need not be shy with regard to hardware capabilities either: Matrox has shown that they can successfully compete with the biggest and the best US companies.

Canadian Prospects

I believe that a very important part of the future of education and training lies with technology-based systems. If Canadians are able to work together in a more cooperative manner than has been the case in the past, I believe that we can produce training products of the highest standards that can play a major role in this industry on a international basis.

The Public Service Commission is dedicated to playing a leadership role in helping to bring this about.

ASSESSMENT OF MARKET POTENTIAL FOR HEALTH RELATED COURSEWARE

Z.L. Hyworon
InfoTech Inc.
Winnipeg, MB

The Government of Canada has identified computer-based training as being of strategic importance to Canadian industry. The Department of Communications commissioned InfoTech Inc. to evaluate the market potential for health related computer-based courseware. As part of this study, a survey was conducted of key Canadian companies, institutions and decision makers whose current and future decisions will shape the market for computer-based training applications.

The survey included the health care sector itself (including hospitals and other health care facilities, organizations or associations), suppliers of products and services, non-profit advocacy groups, as well as industrial users of health and safety training courses and services. The sample was collected through personal and telephone interviews as well as a mailout of 850 questionnaires.

This paper would provide a summary overview of the findings related to present training activities, current use of CBT, management issues and considerations, as well as an overview of priority application areas identified by the participants.

ÉVALUATION DU MARCHÉ POUR LES DIDACTICIELS RELATIFS À LA SANTÉ

Le Gouvernement du Canada a reconnu que la formation assistée par ordinateur est très importante pour l'industrie canadienne. Le Ministère des communications a demandé à InfoTech Inc. d'évaluer le marché des didacticiels relatifs à la santé. Dans le cadre de cette étude, une enquête a été menée auprès de sociétés, d'organismes et de décideurs importants dont les décisions actuelles et futures influeront sur le marché des didacticiels.

L'enquête a été menée dans le secteur de la santé lui-même (les hôpitaux et les organismes et associations de santé), chez les fournisseurs de produits et de services et les groupes de défense sans but lucratif, ainsi que chez les utilisateurs industriels de cours de formation et de services en matière de santé et de sécurité. Les données ont été recueillies par le truchement d'entrevues personnelles et téléphoniques et de 850 questionnaires envoyés par la poste.

La présente communication donne un aperçu des résultats en ce qui concerne les programmes de formation actuels, l'utilisation des logiciels de formation assistée par ordinateur, les questions de gestion ainsi que les domaines d'application prioritaires indiqués par les participants.

ASSESSMENT OF MARKET POTENTIAL FOR HEALTH RELATED COURSEWARE

Zorianna L. Hyworon

President, InfoTech Inc.

The Government of Canada has identified computer based training as being of strategic importance to Canadian industry. This strategic role is seen in the context of meeting the continuing training and educational needs of Canadian industry, new business opportunities for a Canadian courseware industry and export opportunities of products and services in the North American and global markets. The health care sector in Canada represents economic activities of about 8% of the gross provincial product and over $25 billion per annum nationally. Various recent federal and provincial government initiatives have been targeted on the economic development opportunities in the health care sector, with specific emphasis on technology based products and services.

Focusing in on a very specific market niche, health-related computer based courseware, InfoTech Inc. conducted a study to assess the market potential of such products and the opportunities for Manitoba companies to address this market. The study was conducted in the fall of 1988 and was partially funded by the federal Department of Communications under the Canada-Manitoba Communications and Cultural Enterprises Subsidiary Agreement (ERDA). This presentation will provide an overview of the summary findings and will explore some of the market and public policy issues which need to be considered.

TARGET MARKETS

The study considered the potential market for health-related courseware to be broader than the health care delivery sector. Four market sectors were identified for evaluation:

1. Health care delivery institutions/organizations (hospitals, chronic care institutions, clinics, professional associations, health care education institutions and government health agencies with a policy or delivery role)
2. Suppliers of health related products, services, information and training (including pharmaceutical and insurance companies, medical suppliers, health advocacy groups, insurance companies, first aid training organizations)
3. Users of health related training (industry and public sector organizations that could have an interest in health related training)
4. Patient and consumer education (community clinics, government agencies, health professionals)

METHODOLOGY

The primary source of data for this study was a survey sent to over 800 individuals and organizations across Canada, representing the four sectors of the target markets. In addition to the mail surveys, telephone and personal interviews were conducted with 94 individuals. The data gathered from these direct sources was supplemented by research on specific issues identified during the course of the study. The survey was comprised of 198 questions in 4 sections:

1. profile of present training environment
2. present use of computer based training
3. interest in potential off-the-shelf applications
4. considerations related to the use and acquisition of computer based courses

The completion rate for the survey ranged from 15% (mail) to 32% (telephone) and 75% (interview). Some of the respondents chose to provide information in a form other than the questionnaire. With 183 survey participants, the analysis sample contained 173 completed surveys.

PRESENT TRAINING ENVIRONMENT

The present training environment was assessed in terms of:

* the budgetary allocation for training
* the number of employees trained annually
* the number of days of training each received
* the allocation of training budget by types of training, groups of trainees, methods of delivery and sources of training courses
* a subjective assessment of the effectiveness of the current training programs in meeting operational and training objectives
* an assessment of factors which were considered to be constraints in meeting future training needs
* present use of CBT, including source of courses, type of delivery and application areas

Budgetary allocation

Although varying greatly in number of employees and training budgets, all hospitals, regardless of size averaged an annual

expenditure of approximately $100 per employee for training. An exception to this are personal care homes where the average allocation for training was $50 per employee. With the notable exception of organizations where training was directly related to their marketing of health related products or services, industry expenditures for training, on a per employee basis, was similar to that of the health care sector.

Training priorities

Within the health care sector, the highest level of training resources were allocated to management training and professional upgrading, with the lowest priority given to patient education. A notable exception to this were rural hospitals, where the high rate of staff turnover forced employee orientation to the top of the priority list.

Effectiveness of Current Training Programs

On-site, classroom courses were still the predominant format for training, particularly in the health care sector. Although it was recognized that there was a pressing need for alternate forms of course delivery, only 25% of teaching hospitals had implemented self-study courses using video, print or CBT as part of their regular course offerings.

In rating the effectiveness of their current training programs in meeting common training objectives or program delivery considerations, teaching hospitals, large urban hospitals and industry were fairly consistent in their evaluation, rating themselves moderately effective in:

* making training available near the workplace
* meeting demand within budgetary constraints
* minimizing job time loss
* minimizing travel/living costs
* providing timely feedback to the trainee

Similarly, they considered their training programs to be least effective in:

* availability WHEN and WHERE required
* providing timely feedback to the trainer
* tracking trainee performance
* tracking and evaluating course effectiveness
* tracking impact on job performance

Smaller institutions and rural hospitals had greater concerns with budgets and providing feedback to the trainee, trainer and management on the effectiveness of the expenditure of the more limited training resources. In looking to the future, the health care sector respondents ranked budgetary and staff constraints of the highest level of concern to virtually ALL of their organizations. These were closely followed by a growing requirement to demonstrate the effectiveness of training in the competition for budget and staff resources.

Present use of Computer Based Training

Almost half (45%) of all participants in this survey indicated that some form of computer based training was already being used by their organizations.

HEALTH CARE SECTOR	
chronic care facilities	15%
urban community hospitals	33%
rural community hospitals	35%
community clinics	42%
teaching hospitals	57%
SUPPLIER SECTOR	56%
INDUSTRY USER SECTOR	60%

Consistently across all sectors, the predominant use of CBT was for training related to the implementation of computer and information systems (60% up to 90% of CBT usage). Where health care institutions used CBT for professional upgrading, those applications represented only 20% to 40% of their CBT usage. The predominant delivery of CBT courses across all sectors was on microcomputer (4:1 over mainframe courses). In the health care sector, only the teaching and large urban hospitals provided CBT courses on their mainframe or minicomputer systems. All other courses were delivered on micros. Although microcomputers were the training computers of choice in industry as well, the nature of the predominant applications (ie. computer related) and the availability of mainframe capacity and related technical resources were reflected in a higher use of mainframes and dedicated CBT centres.

Further reflecting the primary application of CBT, the majority of the courses offered in this form were acquired from hardware or software suppliers or as commercially available packages. In the health care sector, only teaching hospitals and large urban hospitals indicated any in-house or contract CBT course development. In industry, a much larger percentage of CBT courses were acquired as off-the-shelf courseware, using in business, technical and marketing applications.

Community Clinics

In the health care sector, the community clinics were the only organizations to whom patient education was of a higher priority. This was related directly to the closer links maintained with patients and their immediate community and the counselling and preventative aspects of health care delivery.

Health Profession Education

Institutions involved in the education of health professionals are increasingly turning to self-study as a means of supplementing practical and classroom instruction. The primary medium for such programs is video-tape, with in-house produced as well as

externally acquired courses.

Suppliers of Health Products and Services

For product oriented companies, the effectiveness of sales staff is directly linked with their ongoing training not only in sales techniques but also in the medical and health considerations associated with their products. For suppliers of services, training itself is a strategic product, promoting product differentiation, opening new markets, generating revenue and contributing to the capture of market share. Where training is linked to revenues and market share, there is a much greater investment in training products and technologies. For suppliers of products and services, training is very closely tied to the bottom line.

Industry

The perspectives on health related training in industry organizations are very pragmatic: safety and accident prevention, regulatory compliance and the effective performance of employees as impacted by their physical and emotional health. Health related training is seen in a growing number of large and small organizations as delivering tangible business benefits, protecting their investment in the corporate human resource.

Patient and Consumer Education

The predominant form of patient education remains a one-on-one, patient/health professional exchange which occurs as a consequence of the patient seeking medical attention or intervention. This type of personalized, ad hoc, reactive patient education has placed a very high burden on the health care delivery system. Estimates are that up to 70% of first visits to doctor's offices are for information and reassurance. In response to cost and demand pressures, there is an increased acknowledgement that technology can facilitate public awareness and even serve as a pre-diagnostic to self-diagnostic tool. The "Dr. Oncall" system available through public access terminals and in doctors offices in Montreal takes the public information process to a diagnostic level, offering suggestions for treatment and care.

FUTURE TRAINING REQUIREMENTS

In considering future training requirements, several trend were identified as having an impact on the target markets. These include:

- an increased emphasis on continuing education for health professionals
- an increased awareness of the benefit and need for patient education
- a growing consumer demand for health education
- a growing trend toward the delivery of health care in a non-institutional environment by non-medical caregivers
- the importance of training in the care of aging and elderly for caregivers as well as for the growing geriatric population
- a corporate emphasis on health and wellness programs
- the impact of regulatory requirements for health and safety training
- the increased emphasis on independent self-study
- the growth and changing nature of computerization in the target markets
- the growth and changing nature of the use of computers in training

ACQUISITION CONSIDERATIONS

The potential to derive benefits from the available technology of CBT can be constrained by the myriad technical, organizational and instructional issues facing an organization in its use of a new instructional medium. The technology itself can be very seductive and can divert attention from the end objectives delivering the bottom line benefits. On the other hand, overly cautious low-cost systems may not add sufficient value to the learning process to gain acceptance or to deliver results.

Off-the-shelf courseware has the potential to make the benefits of CBT available to a very broad market at a low cost and with known, previewable quality. Where the subject content is generic and relatively stable, it can be implemented without any additional expense. Where unique requirements exist, the off-the-shelf products can be used as introductory or tutorial materials with in-house instruction focusing on the unique needs.

In considering the potential of CBT courseware in the target markets, it was necessary to assess the perceptions of decision makers with respect to both the positive and negative implications of considering computer based training. Most organizations, across all sectors considered increased and equalized access to training as the most important potential benefit of CBT. This was followed by the improved capability to respond to increasing demands for training and the view of CBT as having the potential enhance learning. All sectors shared in the perception that reducing cost was the least important consideration.

There was similar agreement across all sectors of the perceived constraints to the use of CBT, the major ones being the availability of: dollars, staff and in-house expertise necessary to support in-house initiatives. The only difference between sectors was that the financial constraints of new development were felt most strongly in the health care sector. The least important concerns related to resistance by trainers and trainees as well as concerns about the potential for detailed tracking and reporting of trainee progress and course effectiveness.

When asked to consider various factors that could affect a decision to purchase off-the-shelf courseware, all respondents identified cost as most important followed very closely by the ability to be run on a IBM or compatible PC. Both the factors were rated slightly higher by the health care sector respondents. Least important across all sectors was Macintosh compatibility or a reluctance to use keyboards. In the health care sector, the ability to customize and off-the-shelf course or to add own modules was rated as being of equal importance as cost by teaching and large urban hospitals. The ability to maintain detailed records of progress and formal testing were considered at the second level of importance across all sectors, albeit rated slightly higher by larger institutions. Basic graphics capability was rated relatively higher by the health sector, with advanced graphic less important. Integration of videotape materials was also considered by all sectors to be somewhat more important than videodisc integration. This view supports the high importance previously attributed to the importance of being able to customize off-the-shelf courseware offerings with in-house content.

INTEREST IN SPECIFIC APPLICATIONS

Assuming the availability of quality courseware at a reasonable cost, survey participants were asked to indicate their level of interest in specific applications. General categories of applications directed primarily at the health care sector included: hospital orientation and consent procedures, patient care and nursing procedures, care of the aging and elderly, child care, dental health and nutrition. Applications with a broader interest among suppliers and non-health user organizations in the industrial, service and public sectors included: workplace safety, first aid, wellness, risk factors of major disease, chronic disease management, anatomy and physiology. The high level of write-in suggestions were a further indication of a positive attitude to the use of packaged courseware.

STRUCTURE AND INTERESTS OF TARGET MARKETS

The potential market for health related courseware spans the consumer, corporate and small business sectors in addition to the health care delivery and social service sectors. The various segments of this market, however, have diverse, but often complimentary interests in the development of such courseware, its content and distribution and its use.

Use of courseware

The willingness to use computer based course delivery as part of an overall training program is dependent in the availability of ready, quality packaged courses. With minor exceptions, interest in a "buy" option will not translate itself into an investment in a "make" option unless there is a unique operational need or a strategic advantage to be gained. Potential purchasers and users with an interest in health related courseware include: corporate users of health, safety and wellness applications, small businesses without the corporate health or safety support resources, health care institutions, social and health service organizations and health professionals concerned with patient education.

Distribution of courseware

Within the health care and supplier sectors, there are organizations whose interest in courseware lies primarily in its distribution to their clients in support of their primary marketing or information dissemination activities. These organizations see computer based courseware as a way of extending their market base, improving service to their clients providing for the generation of additional revenues. Such organizations include pharmaceutical companies, insurance companies, advocacy groups, training organizations, government agencies and hospital and professional associations.

Courseware development

There are both public and private sector organizations with a material interest in the courseware being developed in the first place. Their involvement may include providing content, reviewing or endorsing content, sponsoring the development of specific courseware products or even the direct investment in the development of specific strategic products. The potential organizations with such interest are similar to those interested in the distribution of courseware but with a much more pressing and strategic business interest in the availability of specific applications.

GOVERNMENT INTEREST IN TARGET MARKETS

In Canada, government, at all levels, has a strong interest in both the demand and the supply side of the health related courseware markets. On the demand side that interest includes the direct and indirect provision and funding of health and related social services in institutional and non-institutional settings, the definition and enforcement of regulatory requirements related to health and safety, the provision and funding of training programs and in its role as the major employer in all regions of Canada. On the supply side, government interest in the development of courseware is both in having the applications made available and used as well as in the econoric activity and export potential that can be generated by a successful courseware industry.

CONCLUSIONS

There is a strong demand for health related courseware across all target sectors. The impact potential ranges from immediate short-term operational benefits for an individual and institution to long-term systemic benefits across the health care system. However, without access to quality courseware at low, microcomputer software prices, this market may develop much slower than the need and present interest may indicate. Furthermore, it is in the interest of government, from both the demand and supply side of the markets to stimulate the development and use of computer based courseware for health related applications. What is required is a strategic view of stimuli which can be put into play to foster both the development and use of health related courseware across the broad market, generating domestic benefits and export opportunities.

L'EXPÉRIENCE SÉNÉGALAISE D'APPLICATIONS PÉDAGOGIQUES DE L'ORDINATEUR

H.N. Sall
École normale supérieure
Université de Dakar
Dakar, Senegal

Le recours à l'ordinateur et à l'informatique dans les systèmes d'éducation soulève de multiples interrogations dans les P.V.D. En effet, à coté des questions qu'une telle innovation, somme toute très coûteuse, suscite partout quelle que soit la richesse de la nation, les Pays dits en Développement connaissent en plus des débats fort passionnés portant sur l'opportunité de tels investissements. L'actuelle crise économique mondiale et la pauvreté endémique de P.V.D. semblent être des facteurs limitants. Tout comme le marasme économique international malgré de timides relances, la raréfaction des sources de financement est un argument de taille pour les tenants de l'approche prudente. C'est que, également, l'investissement en matière d'éducation ne paraît pas toujours productif ni immédiatement processif.

Or, beaucoup plus que les nations riches, les P.V.D. doivent, sans délai, s'engager dans des processus de modernisation de leur tissu économique et de leur culture (au sens le plus général de ce terme). Jamais l'aide et l'assistance internationales ne remplaceront l'effort et les sacrifices nationaux. A ce titre, l'éducation et l'enseignement ont un rôle essentiel et primordial à y jouer pour gagner les paris sur l'industrialisation et la modernisation.

THE SENEGALESE EXPERIMENT IN PEDAGOGICAL COMPUTER APPLICATIONS

Use of the computer and informatics in school systems prompts many questions in developing nations. In addition to the issues that such an innovation - ultimately very costly - raises in any nation regardless of its wealth, in so-called developing countries, its timeliness is the source of passionate debate. The current world economic crisis and the endemic poverty of developing nations seem to be limiting factors. Just like the international economic slump (in spite of small gains), the decline in funding sources provides supporters of a more cautious approach with a strong argument. More specifically, investing in education does not always appear productive or immediately progressive.

Much more than rich nations, developing countries must, without delay, become involved in modernizing their economies and their cultures (in the most general sense of the word). Never will international aid and assistance replace national effort and sacrifice. Education and teaching therefore have a fundamental role to play if the goals of industrialization and modernization are to be achieved.

L'EXPÉRIENCE SÉNÉGALAISE D'APPLICATIONS PÉDAGOGIQUES DE L'ORDINATEUR

H.N. Sall
École normale supérieure
Université de Dakar
Dakar, Senegal

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

SIMULATION

STRUCTURED AND UNSTRUCTURED APPROACHES TO SIMULATION: WHAT DO THE STUDENTS THINK?

M.V.J. Veenman, J.J. Elshout, D.J. Bierman
Dept. of Psychology, University of Amsterdam
Amsterdam, The Netherlands

The relation between intellectual ability (IQ), novice problem solving behavior and learning in two different learning environments based upon simulation, using AI techniques, in the field of thermodynamics is being investigated using thinking aloud protocols produced by the students. A first instructional approach allows for unstructured, LOGO-type like learning where students have to design some experiments in a simulated heat-lab in order to discover major principles of heat theory. The second approach concerns a more structured learning environment in which students are guided towards understanding major principles of heat theory by conducting pre-set experiments in the heat-lab. Currently, the tape-recorded protocols of eight students, who were selected according to their deficiency in thermodynamics and differed in intellectual ability, have been analyzed. Four of them worked in the structured simulation, the other four in the unstructured version. The results indicated that systematical orderliness (as part of novice problem solving behavior) is the vehicle through which intellectual ability contributes to learning in simulations of knowledge-rich domain like thermodynamics. However, it seems that intellectual ability has some predictive value to the amount of more complex learning. Furthermore the present data do not support strong conclusions regarding the different possible interactions between the structure of the learning environment and the intellectual ability of the students. A second, more elaborate experiment is now being conducted using 24 subjects and a slightly adjusted version of the simulation environments. The main focus of this experiment are the differences in attribution of intellectual ability to either novice problem solving behavior and the amount of what is being learned in structured versus unstructured simulation environments. We think that this kind of elaborate aptitude treatment interaction research will contribute to methods for learning enhancement in the field of CAL.

APPROCHES STRUCTURÉES ET NON STRUÇTURÉES À LA SIMULATION : QU'EN PENSENT LES ÉTUDIANTS?

Le lien entre la capacité intellectuelle (QI), le comportement des débutants pour ce qui a trait à la solution de problèmes et l'apprentissage (dans le domaine de la thermodynamique) par simulation dans deux milieux différents à l'aide des techniques d'IA fait l'objet d'une étude à l'aide des démarches verbalisées des étudiants. La première approche éducative permet un apprentissage non structuré de type LOGO; les étudiants doivent concevoir certaines expériences dans un laboratoire simulé d'étude de la chaleur pour découvrir des principes de base de la théorie de la chaleur. La deuxième façon de procéder fait appel à un milieu d'apprentissage plus structuré: les étudiants apprennent, avec de l'aide, à comprendre les principaux principes de la théorie de la chaleur en effectuant des expériences prédéterminées dans un laboratoire. Actuellement, les protocoles enregistrés sur bande de huit étudiants qui ont été choisis en raison de leur manque de connaissances en thermodynamique et ayant des capacités intellectuelles différentes ont été analysés. Quatre d'entre eux ont travaillé avec le système de simulation structurée alors que les quatre autres ont utilisé la version non structurée. D'après les résultats obtenus, la méthode systématique (comme partie du comportement d'un débutant à l'égard de la solution de problèmes) constitue le véhicule par lequel la capacité intellectuelle contribue à l'apprentissage dans des simulations d'un domaine riche en connaissances comme la thermodynamique. Cependant, il semble que la capacité intellectuelle ait une certaine valeur prophétique pour ce qui a trait à un apprentissage plus complexe. En outre, les données actuelles ne permettent pas de conclure avec certitude au sujet des diverses interactions possibles entre la structure du milieu d'apprentissage et la capacité intellectuelle des étudiants. Une deuxième expérience, plus poussée, est maintenant en cours; on y fait appel à 24 sujets et à une version légèrement modifiée des milieux de simulation. Cette expérience porte principalement sur les différences dans le rôle de la capacité intellectuelle dans le comportement du débutant à l'égard des problèmes et la quantité de choses apprises dans des milieux de simulation structuré et non structuré. Nous croyons que ce type de recherches poussées sur l'interaction dans le traitement des aptitudes contribuera à l'élaboration de méthodes permettant d'améliorer l'apprentissage dans le domaine de l'EAO.

STRUCTURED AND UNSTRUCTURED APPROACHES TO SIMULATION: WHAT DO THE STUDENTS THINK ?

Marcel V.J. Veenman, Jan J. Elshout & Dick J. Bierman

University of Amsterdam.

The relation between intellectual ability (IQ), novice problem solving behavior and learning in realistic, simulated learning environments in the field of thermodynamics is being investigated using thinking aloud protocols produced by the students. Realistic learning environments involve learning in semantically rich domains, in which complex and knowledge-rich problems are representative for everyday problem solving (Jansweijer, 1988). These domains are especially interesting, because the problem settings appeal explicitly to intellectual abilities while being more sensitive to aspects of novice problem solving behavior (poor orientation, untidiness and a loose planning of activities; Elshout, 1987).

Within physics, a semantically rich domain, it is possible to trace explicitly the levels of understanding of the subject matter along the learning process. Take heat theory as an example. Heat theory offers the combination of two distinctive concepts, weight and matter, into one concept: 'specific heat'. Understanding the basic principles of heat theory is related directly to the elaborateness of processing those concepts. It is assumed that integrating these concepts into one quantitative term invokes intelligence, especially logico-mathematical reasoning. Furthermore the concomittants of intelligence, e.g. thorough orientation and systematical orderliness, are likely to facilitate the complex learning process in thermodynamics (Elshout, 1987).

Because of the conceptual complexity of the subject matter, the naive understanding of heat theory is prone to several misconceptions. Frequently encountered misconceptions concern the confusion of heat (energy) and temperature (the property of an object), and the assumption that some materials absorb less warmth than others do (Steehouwer, 1986). These misconceptions might be based on naive everyday experiences.

There are two prevalent approaches in CAL on how to tackle these misconceptions. In the first instructional approach, the learning environment is presented as a LOGO-type 'microworld' (Papert, 1987). This rather unstructured microworld should bear close resemblance to reality in order to avoid interference with the 'natural' learning process (diSessa, 1987). Learning is conceived as a self-regulating exploring process by which major principles of the microworld are discovered.

The second approach is based upon the notion of a 'Socratic dialogue' (Collins & Stevens, 1983). This structured dialogue aims at forcing the student to re-examine his or her conceptions by performing pre-set tasks in a restricted simulated environment. Misconceptions or blatantly inconsistent ideas are explicitly put to the test by the outcomes of these pre-set tasks.

To compare both CAL approaches, two versions of a Heat laboratory simulation (see figure 1.) were implemented in Prolog, extended with an Object-oriented interface (Kamsteeg & Bierman, 1988). The unstructured version requires students to solve some broadly defined problems (e.g. What is the relation between weight, heat and temperature ?) by designing their own experiments in the simulated heat lab. In the structured version students are guided towards understanding major principles of heat theory by conducting pre-set experiments, after being forced to formulate a hypothesis.

Pilot study:

First a pilot study was carried out. Eight students, selected according to their deficiency in thermodynamics and with different intellectual abilities, participated. Four of them worked in the structured simulation, the other four in the unstructured version. In a pretest-posttest design a misconception-questionaire was administered prior to and shortly after the experiment. This questionaire contains a series of statements, which have to be evaluated as right or wrong (e.g. "A certain temperature indicates a certain amout of heat.") Retention was measured three weeks later. The tape-recorded think aloud protocols, produced by the students, were analysed on indications of intelligent

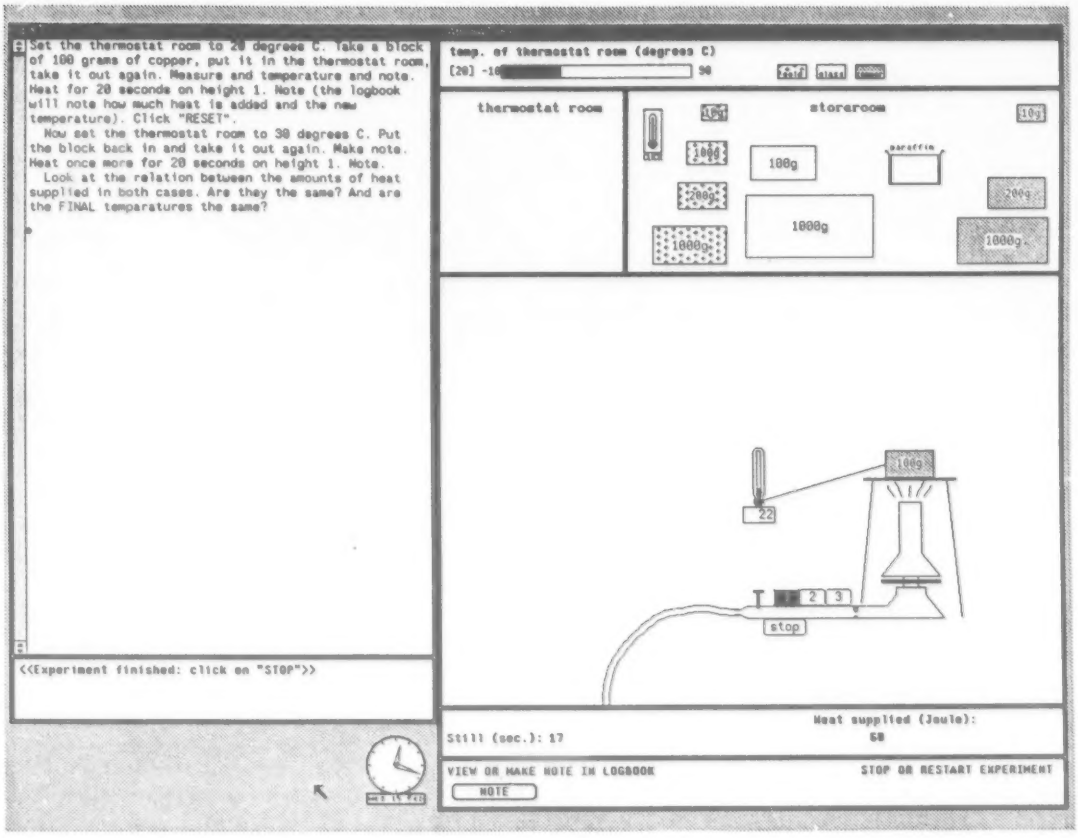


Fig. 1. Screendump of the laboratory.

problem solving behavior, orientation activities and systematical orderliness, and on the level of learning reached. Two judges, who received no information about the student's test scores, performed the analyses together, arguing until they reached agreement. The results indicated that systematical orderliness is the vehicle through which intellectual ability contributes to learning in simulations of a knowledge rich domain like thermodynamics (with R=.89). Intellectual ability (judged intelligence and test scores correlated .81) however, only had a moderately predictive power for the amount of more complex learning. The data did not support strong conclusions regarding the different possible interactions between the structure of the learning environment and the intellectual ability of the students. However, while transcribing the protocols, it turned out that the experimenter intervened unsystematically in order to avoid differences in time-on-task. Some students, especially in the unstructured version, had to be slowed down, while others had to be urged to proceed in time. This may have been responsible for the absence of an Aptitude Treatment Interaction effect and might have reduced learning effects due to differences in intellectual abilities too.

Experiment:

A second, more elaborate experiment was conducted recently, using 26 subjects and a slightly adjusted version of the simulation environments. More effort was spent in persuading subjects in the structured condition to formulate hypotheses explicitly before they were tested in the lab. Time-on-task was not controlled, but was included in the study as a concomitant variable.

Besides a structured and an unstructured condition a third learning condition was included. Subjects in this additional learning condition received a computerized explanation of thermodynamic processes, before entering the structured simulation environment. It was assumed that this explanation would improve performance on the misconception questionaire. The appearance of students jumping from one model of explanation for occurences in the heat-lab to another contradictory model (e.g. using a conductivity model as well as a heat repulsion model),

confirmed the impression that misconceptions are inconsistent explanatory ideas, introduced by a lack of knowledge. Adequate information about what actually happens during heating an object, should fill this knowledge gap.

Method:

Some months prior to the experiment intellectual ability was assessed by a variety of ability tests. Subjects, all first year psychology students, were selected according to a high school deficiency in physics and an high or low compound score on the intelligence tests. The 26 participating students (12 low IQ & 14 high IQ) were randomly assigned to the three learning conditions.

Shuffled versions of the misconception questionaire were administered as pretest and posttest, and as a retention test two weeks afterwards. Thinking aloud protocols of all students were tape-recorded during their 'work' in the heat lab. After completion of the posttest questionaire students were asked to solve three problems while thinking aloud. These problems require a fair amount of quantitative reasoning based upon understanding of heat theory, so they are assumed to measure the depth of processing and the elaborateness of learning in the simulated environments. Answers were rated as either zero (incorrect) or one (correct).

Preliminary results:

No data from the protocol analyses are yet available (they will be forthcoming just before the symposium). The procedures for protocol analysis will correspond to those in the pilot study.

The results reported here, have been obtained by means of a 2 * 3 Anova with intellectual ability (IQ) and learning condition as independent variables. The six groups showed no substantial differences in sexe, age and the amount of previous education in physics. However scores on the misconception pretest showed a significant effect of IQ, $F(1,20)= 6.91$ ($p=.016$), indicating that high IQ subjects are less prone to misconceptions. This effect of IQ also turned up in the misconception posttest scores with $F(1,20)= 10.26$ ($p<.005$). An Anova was performed on the pretest-posttest difference scores, but produced no significant results. A pairwise T-test between pretest and posttest scores indicates that (almost) all of the students improved after training ($T= 6.00$, $df= 25$, $p= .0021$).

The retention test scores again showed a significant effect of IQ, $F(1,20)= 26.09$ ($p= .0001$), but this time the differential scores of pretest and retention test also revealed a significant result of IQ with $F(1,19)= 10.55$ ($p<.01$). Two weeks after the treatment the low IQ subject almost dropped to the level of their pretest scores, whereas high IQ subjects maintained or even improved their level.

In the scores on the 'three quantitative problems' again the IQ factor showed up with a significant result, $F(1,20)= 15.51$ ($p<.001$). High IQ subjects were able to solve about two problems out of three, while low IQ subjects on the average could solve far less than one problem out of three.

Time-on-task differed significantly in every way. First the time-on-task differed between the learning conditions, the structured versions taking about twice as long as the unstructured version, $F(1,20)= 37.15$ ($p=.0001$). Furthermore a significant difference between IQ groups, $F(1,20)= 9.18$ ($p<.007$), arose from the structured versions with low IQ subjects taking more time than high IQ subjects. However, time-on-task correlated low with measures of learning, correlations ranging from -.25 to .10.

Preliminary conclusions:

At the present no learning effects due to differences in learning environment can be detected, nor could an aptitude treatment interaction between intelligence and structure of learning environment be found. However, the significance of intellectual ability for learning in the long run (the retention test) and for more complex learning ('three problems'), is evident. Surprisingly time-on-task was not of any importance.

These preliminary results offer a pessimistic view on education. Whatever you teach in the field of thermodynamics, the smart students take advantage of it, but the less talented relapse into ignorance. The protocol analyses might present us with a deeper understanding of how the more intelligent students benefit from orientation activities and systematical orderliness in a complex learning environment.

Literature:

Collins, A. & Stevens, A.L. (1983) - A cognitive theory of inquiry teaching. In: C.M. Reigeluth (Ed.) *Instructional design theories and models: an overview of their current status*. Hillsdale, Erlbaum.

Elshout, J.J. (1987) - Problem solving and education. In: E. de Corte, H. Lodewijks, R. Parmentier & P. Span (Eds.) *Learning and instruction*. Oxford: Pergamon Books Ltd/Leaven: University Press.

Jansweijer, W.N. (1988) - *PDP: Een benadering vanuit de kunstmatige intelligentie van probleemoplossen en leren door doen in een semantisch rijk domein*. Doctoral thesis, University of Amsterdam.

Kamsteeg, P.A. & Bierman, D.J. (1988) - *A Prolog-based simulation environment for physics as a tool for fundamental educational research*. Paper at the first European Congress Artificial Intelligence and Training, Lille, France.

Papert, S (1987) - Microworlds: transforming education. In: R.W. Lawler & M. Yazdani (Eds.) *Artificial intelligence and education (volume 1): Learning environments & tutoring systems*. Norwood, Ablex Publishing.

Di Sessa, AA (1987) - Artificial worlds and real experience. In: R.W. Lawler & M. Yazdani (Eds.) *Artificial intelligence and education (volume 1): Learning environments & tutoring systems*. Norwood, Ablex Publishing.

Steehouwer, M. (1986) - *Naïeve ideeën en mentale modellen in de thermodynamika*. Internal report, University of Amsterdam.

FYSIODISC - INTELLIGENT INTERACTIVE VIDEOSYSTEM

W. van der Kooij
Institute of Higher Professional Education
Hogeschool van Amsterdam-H.v.A.
Amsterdam, The Netherlands

The Institute of Higher Professional Education, Hogeschool van Amsterdam-HvA in the Netherlands, has been assigned by the Government to explore the topic "Computer Managed Interactive Video Patient Simulation in Physical Therapy". In this field the Sector Health Care developed the "Intelligent Interactive Videosystem", the FYSIODISC.

This FYSIODISC consists of six patient simulations: Physical Therapy, CAI, Interactive Video (Laservision Disc) and a Knowledge-based Computercoach with Expertsystem. The used tools are Goldworks and Flex. The system is developed for an IBM-AT. The used methodology is KADS (Knowledge Acquisition and Structuring). The laservision player is the VP 406 by Philips Holland. With these kinds of simulations an end was put to educational limitations concerning the subjects "Examination and Treatment" within the study Physical Therapy: a) patients with the desired characteristics are always readily available; b) students are no longer afraid to make mistakes; c) wrongly chosen examinations or treatment have no consequences for the patient; d) no more invasion of the patients privacy; and e) the simulations are "real life" situations.

By directing these patient simulations on certain aspects of reality, certain skills and knowledge areas can be extensively practised. The Knowledged-based Computercoach with Expertsystem gives this product a strongly innovative character (High Tech.).

FYSIODISC - SYSTÈME INTERACTIF AVEC INTELLIGENCE ARTIFICIELLE

L'école de Formation Professionnelle Supéreure d'Amsterdam-H.v.A. a été indiquée par le Gouvernement des Pays-Bas pour explorer les possibilités d'enseignement sur les professions paramedicales par ordinateur, le soit dit "Computer Managed Instruction" (Instruction menagée par ordinateur). A ce propos le sécteur de Santé de cette école a développé le "FYSIODISC".

Le "FYSIODISC" contient six simulations de patient en physiothérapie assisté par ordinateur en interaction avec laservision et un système d'expert (intelligence artificielle). Le système est à base de GOLDWORKS et FLEX et a été développé pour des machines compatibles à IBM-AT. La méthodologie utilisées est celle de KADS (Knowledge Acquisition and Structuring). Le joueur laservision est le VP 406-Philips Hollande. Grâce à cette forme de simulation les limitations dans l'enseignement en matières (para)-médicales comme "le diagnostic et la thérapie" peuvent être supprimées: a) des exemples de patients avec la pathologie désirée sont facilement disponibles; b)les étudiants peuvent se permettre des fautes sans conséquences pour le patient; c) il n'y a plus d'interventions sur la vie privée des malades; et d) les simulations sont et restent de situations de pratique.

De cette façon l'étudiant a la possibilité d'éxercer sa connaissance et sa pratique fréquemment. L'utilisation d'un système d'expert donne à ce produit un caractère très innovatif.

FYSIODISC - INTELLIGENT INTERACTIVE VIDEOSYSTEM

W. van der Kooij

Institute of Higher Professional Education
"Hogeschool van Amsterdam - HvA", The Netherlands
Sector Health Care (Paramedics & Nursing)

1. Introduction:

The Institute of Higher Professional Education "Hogeschool van Amsterdam - HvA", has been assigned by the Dutch Government to explore the topic "Computer Managed Interactive Video Patient Simulation in Physical Therapy".
The Sector Health Care of the "Hogeschool van Amsterdam - HvA" developed the "Intelligent Interactive Videosystem", the FYSIODISC, with financial support from the Ministry of Education & Science, the Ministry of Economic Affairs and from participating industries (Bolesian Systems Europe, Helmond; Media Design, Amsterdam; Philips Eindhoven; IBM, Amsterdam) and institutions (University of Amsterdam; Free University Amsterdam).

2. The FYSIODISC program:

The FYSIODISC consist of six patient simulations physical therapy, C.A.I., interactive video (Laser Vision Disc) and a knowledged-based computercoach (A.I.).
The knowledged-based computercoach gives the FYSIODISC a strongly innovative character.
With the FYSIODISC program the student is taught a method for solving physical therapeutic problems. The student is instructed to first generate rather general hypotheses on the basis of the data from the history taking process and the inspection. These general hypotheses must be specified and verified by selecting the correct physical examinations. Eventually only one specific hypothesis (the physical therapeutic diagnosis) will remain.

For learning the problem solving skills a form of Computer Aided Instruction (C.A.I.) was selected, with which the student acquires the most immediate and specific form of feedback regarding the problem solving process followed by him.
To realize this, an Intelligent Tutoring System (I.T.S.) is employed. This I.T.S. closely follows the problem solving process of the student and, by means of a dialogue with the student, it can establish why the student has arrived at a certain choice, and it can provide a feedback. At the same time the I.T.S. gives us the possibility to lay down the shortcoming of knowledge by students. It is for example possible to recover the systematically made mistakes by students.

With the FYSIODISC program the student may pose questions to the patient and carry out examinations. The answer to the question are given in the form of a computer text and the student can study the examinations on the video monitor.
The images originate from the Laser Vision Disc.
The choosen problems are called "cases". The level of the "cases" are suitable for the training of physical therapy students of the second and the third semester.
The "cases" of the FYSIODISC are:
- three "cases" of the upper extremity;
- three "cases" of the lower extremity.

The FYSIODISC program provides a chance to the student to practice the physical therapeutic problem solving process in a structured manner whereby:
- patients are always available;
- any errors by the student do not have adverse consequenses for the patient;
- students need not feel a sense of failure;
- the simulations are "real life" situations.

3. The structure of the FYSIODISC program:

1. Introduction of the patient;
2. History taking process;
3. Physical examinations;
4. Supplementary medical data;
5. Formulation of the physical therapeutic problem;
6. Formulation of treatment & planning.

4. The design of the I.T.S.:

An I.T.S. is build with at least the following four components:
- a domainexpert;
- a student module;
- a didactic module;
- an interface.

Within this I.T.S. te didactic module has two parts:
- a diagnosticus;
- a coach.

The domainexpert is capable of performing the physical therapeutic diagnosis.
The student module presents the knowledge of the student.

The diagnosticus has the task to detect the wrongly performed actions of the student and to determine what was wrongly performed by the student.
The coach takes care of the wrongly performed actions of the student, diagnoses these actions and gives the student the desired feedback.

5. The architecture of the I.T.S.

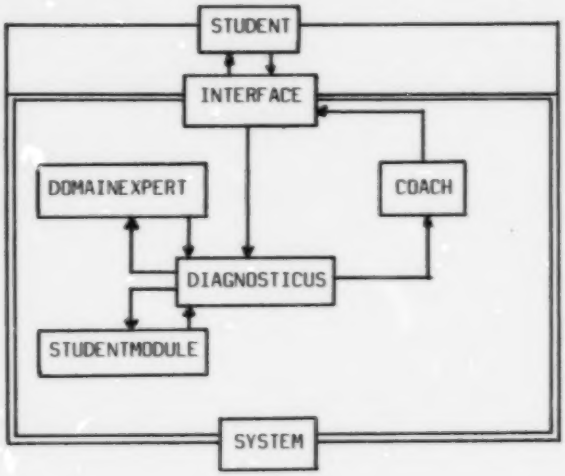


fig. 1

6. Equipment configuration:

Operating System & Computer:
The software is for personal computers running the MS-DOS Operating System.
(Version 2.1. or higher).
The prefered PC is an IBM PC AT or a 100% compatible and the IBM PS/2.

Laser Vision Player:
The driving commands require the use of a Laser Vision Player from the 400 series by Philips.

The FYSIODISC program makes use of 2 monitors, a computermonitor and a videomonitor.

The product FYSIODISC consists of a Laser Vision Disc, diskettes and manuals for teachers and students.
The product is to be purchased from:
Institute of Higher Professional Education
"Hogeschool van Amsterdam - HvA", Sector Health Care (Paramedics & Nursing)
Ter Haarstraat 17, 1053 LH AMSTERDAM,
The Netherlands. Phone: (0)20-122111.

SIMULATION OF ENGINEERING MODELS BY INTERACTIVE COMPUTER GRAPHICS

Y.P. Lee, C.S. Chang
Dept. of Computer Science
The Chinese University of Hong Kong
Shatin, Hong Kong

Many engineering concepts can now be taught in a quite simple way by using interactive computer graphics. The model may be uniquely represented in terms of a partial differential equation. The student will first get to know the model by using just one physical concept which he is familiar with. The student can have a real time response in the monitor for the change of input parameters he made for the boundaries. Once the student has mastered the concept of the model, he will find no difficulty to understand many other engineering concepts belonging to the same model by interactive with a choice from a menu on the monitor.

Techniques used to implement this program are outlined in the paper. A SUN 3/110 workstation was used to implement the program.

SIMULATION DE MODÈLES TECHNIQUES AU MOYEN DE L'INFOGRAPHIE INTERACTIVE

Actuellement, il est possible grâce à l'infographie interactive, d'enseigner de nombreux concepts techniques d'une manière simple. Le modèle peut être représenté à l'aide d'une équation aux dérivées partielles. L'étudiant se familiarisa d'abord avec le modèle en utilisant un seul concept physique qu'il connaît bien. Il pourra voir à l'écran une réponse en temps réel aux changements des paramètres d'entrée qu'il a effectués aux limites. Lorsque l'étudiant connaîtra à fond le concept du modèle, il comprendra facilement beaucoup d'autres concepts techniques du même modèle, en dialoguant avec une option du menu à l'écran.

Les techniques utilisées pour mettre en oeuvre ce programme sont résumées dans la communication. Le poste de travail utilisé était un SUN3/110.

SIMULATION OF ENGINEERING MODELS BY INTERACTIVE COMPUTER GRAPHICS

Y.P. Lee and C.S. Chang

Department of Computer Science
The Chinese University of Hong Kong
Shatin, Hong Kong

1. INTRODUCTION

Computers are now used widely in engineering education related to teaching undergraduate courses. Computer simulation also plays an important role in the study of engineering phenomena for which adequate mathematical models are available[1]. Unfortunately, because of its mathematical nature, the fundamental concept is often considered to be difficult for students to understand.

Traditional engineering education will involve experiments to let the student have a better understanding of the physical phenomena. However, in many engineering experiments, it is very expensive to set up the laboratory, the emergence of interactive computer graphics provides an alternate or sometimes even a better way to conduct engineering experiments through graphical simulation[2]. Computer graphics helps students to understand more clearly about the complex physical concepts associated with many engineering disciplines[3]. In a highly interactive environment, students can react with the system by requesting various pieces of information, and or by varying the input parameters in order to study their effects on the characteristics of the system. As a result, interactive graphical simulation is now recognized as a new tool in the teaching of engineering courses.

Among all the mathematical models, Laplace equation is the simplest type and the most generalized one because all the diffusion problems are governed by this single second-order partial differential equation. For example, thermal conduction in a solid or fluid medium and neutron diffusion in a nuclear reactor. There are so many engineering problems which may be modelled by the Laplace equation, hence successful simulation of it will be a close analogy to the reality. In this paper, we will consider only the steady-state heat flow in a rectangular region as a feasibility study on simulation of engineering models by interactive computer graphics.

2. BACKGROUND OF TRADITIONAL METHODS TO SOLVE THE LAPLACE EQUATION [4]

2.1 Free-Hand Sketching

In a two-dimensional system the flux lines and equipotential lines form an orthogonal set of curves. A very good estimate of a two-dimensional field pattern may be obtained by free-hand sketching flux and equipotential lines. However, it is essentially an approximate method with 10 per cent accuracy and is very time consuming when done manually.

2.2 Experimental Techniques

The methods of experimental field plotting involve the using of electrolytic tank, rubber membranes, soap film, Hele-Shaw fluid-flow equipment and photo-elasticity. This approach mandates the use of equipment that is expensive to purchase and maintain, and that may be used only a few times a year. Moreover, changing the boundary conditions may require to set up the experiment again, which is not time and cost effective.

2.3 Mathematical Analysis

The analysis may employ images, conjugate functions, conformal transformation, separation of variables, cylindrical and spherical harmonics. Typically, conformal transformation is by far the most powerful method, being capable of handling boundaries of much more complicated shape than other analytical methods[5]. It can be used to analyse the field and in many cases, allow the direct calculation of field maps. The chief limitation in applying transformation techniques, is that, for most problems, the appropriate equation of transformation must be known, boundaries have to be assumed to be equipotentials, or coincident with flux lines, or combination of these two types.

2.4 Numerical Methods

Numerical approximation employing digital computer, will give the quickest and most accurate results. Often the numerical solution of the differential equations has the advantage of being applicable to problems that have no analytical solution. The numerical method most often used is that employing finite differences.

3. GRAPHICAL INTERPRETATION OF LAPLACE EQUATION

It is well known that the Laplace equation is of the form:

$$\frac{\partial^2 t}{\partial x^2}+\frac{\partial^2 t}{\partial y^2}=0$$

Students may be very familiar with the derivation of Laplace equation, but they may not understand its physical concept.

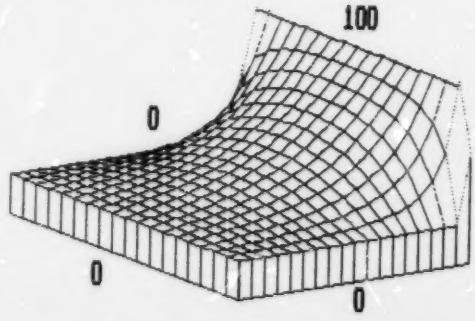


Figure 1 A surface of temperature distribution

Consider figure 1, it is shown a surface of temperature distribution for a two dimensional steady-state heat flow in a rectangular region. With the aid of interactive computer graphics, the students may interact with different boundary values and will notice that the temperature surface generated is unique, because for any point on the surface, temperatures at neighbouring points bear a unique relation with the temperature at that point according to the Laplace equation. If we interpret a few equally spaced points within the region as shown in figure 2(a), we may find that the terms

$$\partial^2 t/\partial x^2 \text{ and } \partial^2 t/\partial y^2$$

define the rate of change of temperature gradients in the x- and y-directions respectively, with respect to the centre point (x_i,y_j).

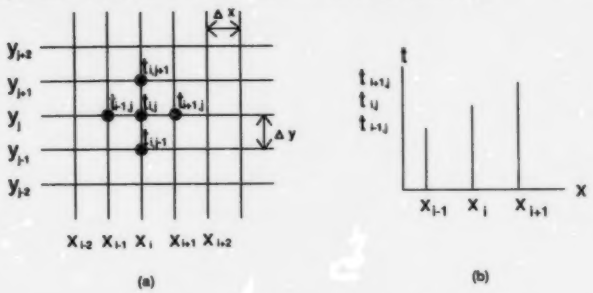


Figure 2 (a) Equally spaced points on a surface of temperature distribution

(b) Section view of temperature at y = constant

As in figure 2(b), suppose the average temperature gradient for the region between $x = x_{i-1}$ and $x = x_i$ is:

$$\frac{\partial t}{\partial x} = \frac{t_{i,j} - t_{i-1,j}}{\Delta x}$$

The rate of change of temperature gradients for the region from x_{i-1} to x_{i+1} will be:

$$\frac{\partial}{\partial x}\left(\frac{\partial t}{\partial x}\right) = \frac{\left(\frac{t_{i+1,j} - t_{i,j}}{\Delta x} - \frac{t_{i,j} - t_{i-1,j}}{\Delta x}\right)}{\Delta x}$$

centered at $x = x_i$.

Similarly, the rate of change of temperature gradient in y-direction at $y = y_j$ is:

$$\frac{\partial}{\partial y}\left(\frac{\partial t}{\partial y}\right) = \frac{t_{i,j+1} - 2t_{i,j} + t_{i,j-1}}{\Delta y^2}$$

Hence, if $\Delta x = \Delta y$, the Laplace equation becomes:

$$\frac{1}{\Delta x^2}(t_{i-1,j} + t_{i+1,j} + t_{i,j-1} + t_{i,j+1} - 4t_{i,j}) = 0$$

Therefore for any point (x_i, y_j) on the surface, the temperatures at its four neighbouring points must satisfy the above equation. Although the Laplace equation has been interpreted with reference to heat flow, it applies equally well to steady-state diffusion problems and to electrical-potential distribution[4].

4. PROGRAM DESCRIPTION AND FEATURES

As the program is a preliminary study to develop techniques for interactive graphical simulation of engineering models, therefore its feature at present is limited to solving the Laplace equation in a rectangular region only. Basically, the program comprises three subwindows, namely, text subwindow, panel area and terminal emulator (tty) subwindow, where each has its own functions described as follows:

4.1 Text Subwindow

The basic function of a text subwindow is to display the on-line help and the tutorial about Laplace equation. The on-line help provides the student a brief description on how to use the system, separate help message available at different level within the system. The tutorial, aided with some graphics diagrams, is divided into three sections: the first section gives an introduction to the Laplace equation; the second and the third sections will teach the students how to solve the equation by the separation of variables method and the finite difference method respectively. Text within the subwindow can be scrolled up or down, this facilitate the student to move to the section wherever he/she wanted. Moreover, the tutorial can be terminated at anytime.

4.2 Panel Area

The panel area contains the items through which the student interacts with the system (i.e. the data input area). Like the help facility, all the input items on the screen can be selected with a mouse attached to the SUN-3/110. Typically, a pull-down menu is provided for the students to define the length and the width of the rectangular region. The temperature at the boundaries are also specified by the students through the use of a slider on the screen.

4.3 Terminal Emulator Subwindow

The terminal emulator (tty) subwindow emulates a standard Sun terminal. Thus, the graphical simulation program and the tutorial diagrams are displayed in this tty subwindow.

5. IMPLEMENTATION

The graphical program for simulating the steady-state heat flow in a rectangular region is implemented by the SunCore graphics package on a SUN-3/110 workstation. The programming language used is the C language.

5.1 Solving Laplace Equation By Finite Difference Method

In all kinds of partial differential equations, the derivatives can be represented by difference quotients. This is equivalent to convert a differential equation into a difference equation. For a steady-state heat flow in a rectangular region, the region can be replaced by a gridwork where each point at the intersection of gridwork is a node. Each node together with its four neighbouring nodes will satisfy the corresponding difference equation. By solving these equations for all the nodes with known boundary conditions simultaneously, the temperature at each node can be determined. In general, the central difference method is used to replace the derivatives instead of the forward or backward difference methods since the error involved in the central difference method is much less and is of the order of $O(h^3)$, where h is the interval length[6].

For any node (x_i,y_j)in a gridwork, if all the nodes are equally spaced, i.e. $\Delta x = \Delta y = h$, the Laplace difference equation is simplified to

$$\frac{1}{h^2}[t_{i+1,j} + t_{i-1,j} + t_{i,j+1} + t_{i,j-1} - 4t_{i,j}] = 0$$

5.2 Iterative Techniques

To solve the above simultaneous difference equations, large amount of memory and computation time are required. Eventually, several iterative techniques are available to solve the stated problems. Through the comparison of memory requirement and execution complexity, it is found that the successive over-relaxation (S.O.R.) method is the best[7].

5.3 Graphical Display Techniques

We have solved the Laplace difference equation by the S.O.R. method, the temperature at each node is represented by a colour correspond to that value. However, the temperature distribution within the rectangular region cannot be shown clearly with such limited number of discrete pixels as shown in figure 3.

In principle, the ideal and most accurate approach is to compute the temperature within the region point by point. However, it violates two most important constraints : a very large memory requirements and an enormous computing time.

In order to avoid the problems stated above, we use the bicubic Bezier surface method to generated the surface of temperature distribution based on those limited number of known values[8]. The region is divided into sub-regions, called patches, as shown in figure 3, each of which is defined

by 16 control points (4x4).

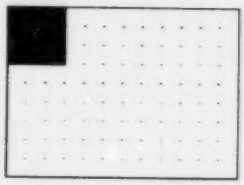


Figure 3 Temperature represented by pixels and a patch in the region

Although the bicubic Bezier surface method is used to calculate the temperatures, we cannot define the size of the rectangular region arbitrary. Suppose the interval length between equally spaced points is h pixels, since the number of patches used to approximate the region must be an integer and each patch is composed of 16 points (i.e. the size of a patch is 3h x 3h), then the length or the width of a region must be divisible by 3h and the result is the number of patches per row or column. To fulfil this requirement, h is set to 20 pixels and the length or width is a multiple of 60 pixels.

5.4 Plotting of Equipotential Lines

In a raster-scan display system, the equipotential lines, i.e. the temperature contour lines for the heat flow problems, can be easily displayed because the temperature values are indicated by different colours. Since the colour index is restricted to integers in SunCore[9], the calculated temperature values within a certain tolerance is forced to map to the nearest integer value. As a result, a equipotential line of x degree in a raster-scan display is actually represented by a narrow band of discrete pixels within a tolerance of, say, ±0.5 degrees as illustrate in figure 4.

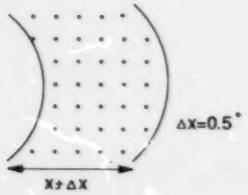


Figure 4 An equipotential band represent x±0.5 degree

5.5 Plotting of Flow Lines

Because of the discrete nature of the equipotential band, it is not easy tc plot the flow lines in a raster-scan display. In order to ensure the orthogonality of the flow line, the normal vector approximation method is attempted[7], in which, two pixels with nearly potentials are used to define the slope of the tangent and the flow line is drawn from a pixel which is made to be perpendicular to the tangent of the equipotential as shown in figure 5.

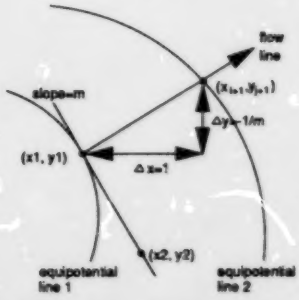


Figure 5 The relationship of the tangent and its normal between two equipotential lines

Mathematically, the slope for two points (x_1,y_1) and (x_2,y_2) is:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad \text{.....(1)}$$

and the slope of its normal, representing the flow line, is:

$$\overline{m} = -\frac{1}{m} \quad \text{.....(2)}$$

Combining equations (1) and (2), the following relationship is obtained which is position for the next theoretical point on the flow line:

$$\Delta x = 1 \text{ pixel}\ ,\ \Delta y = -\frac{1}{m} \text{ pixel} \quad \text{.....(3)}$$

The next pixel on the flow line is determined by equation (3), which is on the basis of incrementing one pixel at a time in x-direction. Thus, the flow line is represented by pixels orthogonal to the equipotential band.

5.6 The Interface

The interface is implemented by SunView which is a system to support interactive, graphics-based application running within windows, all features provided in SunView can be easily manipulated using C language[10].

6. ILLUSTRATIVE EXAMPLES

To illustrate the various features of the programs, some screen prints of the graphics displays have been obtained for typical examples, as described below.

Example 1 - Figure 6(a) shows the input buttons to initiate commands. Figure 6(b) shows the pull-down menu to define the length of a region. Figure 6(c) shows the slider which is used to control the boundary temperature.

Example 2 - Figure 7 shows the graphical simulation of steady-state heat flow in a rectangular region and also the heat flow lines.

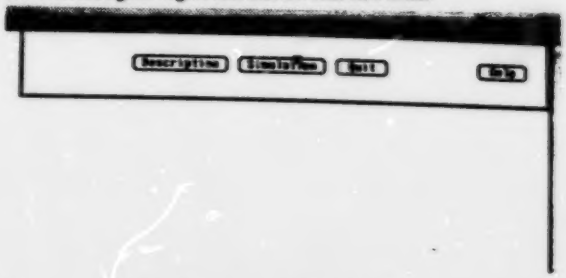


Figure 6(a) Example of input buttons

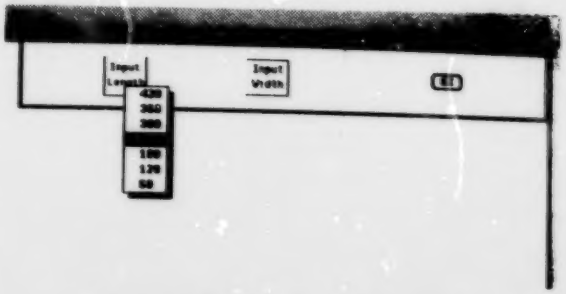


Figure 6(b) Example of a pull-down menu

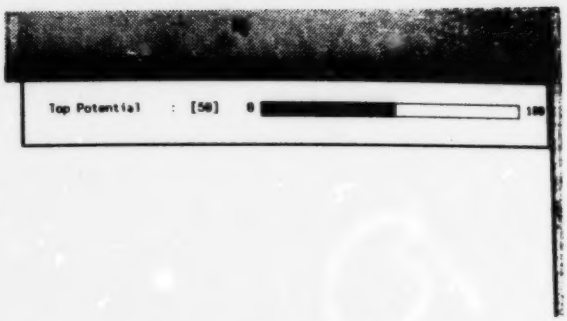


Figure 6(c) Example of a slider

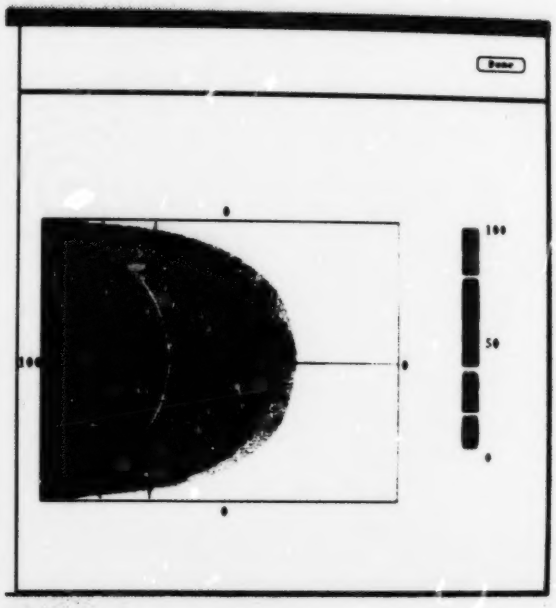


Figure 7 Steady-state heat flow and its flow lines

7. CONCLUSIONS

The result of the preliminary study on the techniques required for the interactive graphical simulation of engineering model for a field problem is presented in this paper. The use of computer graphics to represent engineering concepts is justified.

In view of the continuing of popularity of the programs, several items remains to be attended to before this project may be considered satisfactory. For example, to handle the case where the boundary is irregular, to implement the graphical simulation for the unsteady-state case where the boundary condition vary with time and finally, to make the system applicable to other partial differential equations (eg. Poisson's equation) in order to quantify the pedagogical value on engineering curriculum.

ACKNOWLEDGEMENT

Figure 1 was plotted by a program written by Mr. T.Y. Cheng.

REFERENCES

1. Ranaweera, M. P., Resman M. F. and Twickler R. G., *An Application of Interactive Computer Graphics for Simulation of Vibratory Systems*, Computer and Education, Vol. 6, pp. 373-379 (1982).
2. Pugh, G. A. and Libii, J. N., *Computer Graphics in Fluid Mechanics Education*, 1986 Frontiers in Education Conference Proceedings, pp.24-27 (1986).
3. Wozny, M. J., *Key Issues in Instructional Computer Graphics*, Computer and Education, Vol. 5, pp. 183-192 (1981).
4. Baden Fuller, A.J., *Engineering Field Theory*, Pergamon Press (1973).
5. Binns, K. J. and Lawrenson, P. J., *Analysis and Computation of Electric and Magnetic Field Problems*, Pergamon Press (1973).
6. Gerald, C. F., *Applied Numerical Analysis*, Addison-Wesley publishing company (1980).
7. Lee, Kenneth Y.P., *Interactive Graphical Simulation of Laplace Equation*, Internal Paper of The Chinese University of Hong Kong (1988).
8. Foley, J. D. and Van Dam, A., *Fundamentals of Interactive Computer Graphics*, Addison-Wesley publishing company (1984).
9. *SunCore Reference Menu*, Sun Microsystems, Inc. (1986).
10. *SunView Programmer's Guide*, Sun Microsystems, Inc. (1986).

IMPLICATIONS OF COMPUTER SIMULATIONS IN SPECIAL EDUCATION

M. Hollingsworth
Rehabilitation Studies
University of Calgary
Calgary, AB

This paper will present the results of an experimental study that was conducted to determine the effectiveness of teaching an explicit strategy to special education students for solving a series of health problems using computer simulations. A pretest was administered to all twenty students to determine entry level health knowledge. Students were randomly assigned to one of two instructional groups. Students in the treatment group received conventional health instruction, practice in decision making using a computer health simulation, and an explicit strategy. The control group received the same instruction, but did not receive an explicit strategy. The computer simulation enabled students to practice applying knowledge gained through conventional instruction across a range of computer simulated health profiles. After two weeks of instruction students were tested and interviewed to determine four different factors: a) generalization of knowledge to videotaped health profiles, b) performance on test computer simulations, c) gains in knowledge through this instructional combination, and d) attitudes toward this form of instruction. The results indicated a significant difference between groups on one of the videotape measures, indicating students taught an explicit strategy were more likely to correctly identify required health changes. Students in the treatment group outperformed the control group on four of the six test computer simulations. All students showed significant learning of health information over the period of instruction. Students indicated a strong preference for using the computer and the simulation; few wanted to continue studying health using traditional texts. The paper concludes with a discussion of three important implications of the results: a) the teacher's role when using a computer simulation, b) the effect on student attitude, and c) the effect on student learning.

RÉSULTATS DE L'UTILISATION DE SIMULATIONS SUR ORDINATEUR DANS UN ENSEIGNEMENT SPÉCIALISÉ

Cette communication présente les résultats d'une étude expérimentale qui avait pour but de déterminer l'efficacité des simulations sur ordinateur pour enseigner une stratégie explicite visant à résoudre des problèmes dans le domaine de la santé. On a fait passer un test préliminaire à 20 étudiants pour déterminer leurs connaissances en matière de santé. Les étudiants ont été divisés en deux groupes de façon aléatoire. Dans le premier groupe, la formation était constituée d'un enseignement pratique, d'exercices pratiques en prise de décisions au moyen de simulations sur ordinateur et d'une stratégie explicite. Le deuxième groupe a reçu la même formation, sans la stratégie explicite toutefois. La simulation sur ordinateur a permis aux étudiants de s'exercer à appliquer les connaissances acquises de la façon traditionnelle à une gamme étendue de profils de santé simulés sur ordinateur. Après deux semaines de formation, les étudiants ont été testés et interviewés pour déterminer quatre paramètres différents : a) l'application des connaissances aux profils de santé enregistrés sur bande vidéo, b) la performance dans les simulations sur ordinateur, c) les connaissances acquises via cette combinaison de méthodes d'enseignement et d) les attitudes face à ce type d'enseignement. On a constaté une différence importante entre les groupes dans l'un des tests: les étudiants disposant d'une stratégie explicite ont pu identifier plus exactement les changements nécessaires. Les étudiants du premier groupe ont mieux réussi que ceux du second groupe dans quatre des six simulations sur ordinateur. On a constaté que tous les étudiants avaient acquis de nouvelles connaissances de façon substantielle au cours de cette expérience. Ils avaient une préférence marquée pour l'ordinateur et la simulation; ceux qui préféraient l'enseignement traditionnel étaient peu nombreux. La communication se termine par une discussion de trois aspects importants : a) le rôle de l'enseignant lorsque l'enseignement se fait par simulation sur ordinateur, b) l'effet sur l'attitude de l'étudiant et c) l'effet sur l'acquisition des connaissances.

IMPLICATIONS OF COMPUTER SIMULATIONS IN SPECIAL EDUCATION

M. Hollingsworth

Rehabilitation Studies, University of Calgary
Calgary, Alberta

INTRODUCTION

New approaches to teaching health are now possible. Computer simulations offer the opportunity to examine health instruction within a new framework. Similar to advanced simulation work offered in the medical field (Jolly, 1982; Norman, Muzzin, Williams, & Swanson, 1985), precollege students can now begin exploring decision making in health instruction through the use of microcomputers.

The need for such instruction obviously exists. Coombs (1986) indicates that in the past most illness was caused by disease. Today, however, "the major causes of ill-health are related to lifestyle - smoking, stress, alcohol and drugs, and lack of fitness." Coombs argues that through effective health education and appropriate changes in lifestyle, the general population's health status could be improved and health care costs could be significantly reduced. Precollege education that directs its energy toward making students cognizant of healthy lifestyles and engages students in making health benefiting decisions would be of obvious long-term benefit.

The Health Ways computer simulation and accompanying materials (Carnine, Lang, & Wong, 1983) focus specifically on these kinds of health issues. Historically precollege students have been able to examine and discuss the impact of decision making in the health area, but have lacked the ability to intensively practice making health decisions. The Health Ways package offers students not only the opportunity to learn about the content of health instruction through traditional classroom experiences, but also provides students the opportunity to practice the process of effective decision making by using a computer simulation.

Woodward, Carnine and Gersten (1988) researched the effect of using a computer simulation and an explicit decision making strategy in teaching health to a special education population. Thirty mildly handicapped senior high school students were randomly assigned to one of two instructional groups. In the control group students were instructed using structured teaching combined with traditional extension exercises and paper-pencil application exercises. The treatment group received structured teaching, the Health Ways computer simulation, and were provided with an explicit decision making strategy to apply to the computer simulation. Instructional time was held constant across the two groups.

Following twelve days of instruction, students received two different posttests. The first test was a conventional, fill-in-the-blank, knowledge test that was also given as a pretest. The second measure consisted of a set of three written health profiles, similar to the information presented in the computer simulation. It measured students' ability to problem solve by identifying specific health problems and making correlated lifestyle changes.

Two major findings were revealed. First, it was found that the treatment group performed significantly better on the subcomponent of the knowledge test that was reinforced by the health simulation. Second, the treatment group outperformed the control group on problem solving abilities. Students in the simulation group were significantly better at identifying and prioritizing specific health habits that needed changed.

RESEARCH QUESTIONS

The recent introduction of computer simulations to precollege classrooms leaves many questions, about instructionally effective methods of using simulations, unanswered. Research by Woodward et al. suggests the combination of simulation and conventional instructional methods is more effective than conventional instruction alone. However, questions remain about the best method of implementing computer simulations in the classroom and the effect of implementing such simulations. The specific questions investigated in the current study were:

1. What is the level of student performance on computer simulations?
2. What gains in student knowledge are realized when taught through a combination of computer simulation and conventional instruction?
3. Can students generalize knowledge gained from the computer simulation and conventional instruction to other contexts?
4. What are student attitudes toward this form of instruction?

The first research question, described above, enabled the researcher to determine the effect of teaching an explicit strategy on students' ability to perform on the computer simulation. The second question centered upon students' learning of health facts and concepts. The third research question was a focal point of this investigation. Students do indeed make learning gains through the combination of computer simulation and conventional instruction, however they may be unable to transfer this knowledge to contexts outside the computing environment. This is an important question to be answered for education in general and is of particular concern within the field of special education. This topic will be examined at some length. The fourth question addressed student perceptions of working on the computer simulation, working in traditional instructional settings, and perceptions of learning gains.

BACKGROUND

Knowledge transfer or generalization is of common concern to nearly all areas of education. The purpose of education is not solely for learning or performance in a

single classroom setting. Rather the intent is to ensure that knowledge and skills taught in the classroom will be utilized in situations outside the instructional setting. As Clark and Voogel (1985) point out, the purpose of training is to provide skills for future benefit to students.

Knowledge transfer is not a guaranteed or automatic process. Certain instructional elements can affect the level of transfer. Stokes and Baer (1977) did an indepth review of 120 studies that have contributed directly to the development of theories of generalization. Based upon this review the authors proposed nine different instructional techniques that affect knowledge generalization. These techniques include accounting for common stimuli, basing the number of exemplars upon the level of the learner, and training that is sufficiently "loose" as to provide a breadth of stimuli.

Engelmann and Carnine (1982) suggest that behavioral theory does not provide a basis for accommodating generalized learning. They argue that in addition to behavioral analysis, developers of effective instruction must also analyze knowledge systems and the act of communication. Through appropriate analysis of instructional content and analysis of the communication of this information to the learner, effective instruction may occur. Generalization of such learned knowledge should occur naturally through effective development of the instructional set.

Although the literature reflects a wide range of variables that may affect knowledge generalization, one commonality does pervade the literature. Independent of the learner population, it is important that instruction be well designed. One means of accounting for well designed instruction leading to knowledge generalization is through careful analysis of events occurring in the generalized setting, then selecting training examples that sample the range of relevant events and response variations (Horner, Jones, & Williams, 1985)

The computer simulation accounts for the design and selection of exemplars by providing a broad range of health profiles with a variety of levels of complexity. These profiles provide students with the opportunity to experience a set of health and lifestyle problems that represent the kinds of health problems one would expect to see in the "real world".

Although knowledge generalization has received considerable attention in the literature, little research has been conducted on the generalization of knowledge from computer environments. A number of researchers (Clark, 1985; Salomon, 1984; Roblyer 1985) have identified the lack of research and urgent need for study in the area of knowledge generalization from computer-mediated instruction. Computer simulations in the classroom encourage the teaching of models that already exist outside the computer environment. The ability to generalize from these simulated models to the real environment is absolutely necessary if computer simulations are to serve a purpose in our educational system. The current study examined the effect of providing an explicit problem solving strategy on the generalization of knowledge from computer simulated environments to other noncomputer environments.

DESIGN AND METHODOLOGY OF THE STUDY

The study involved twenty-one junior and senior high school students that were currently receiving special educational services. A reading comprehension test was administered to all students, yielding an average reading ability of grade 5.5. Students were randomly assigned, through matched pairing based upon the reading test, to one of two groups.

All students received four days of preliminary instruction. This initial orientation assisted students in acquiring the basic skills required to use the health simulation effectively. Students were introduced through structured teaching and simulation tutorials to basic vocabulary words, to the simulation objective, and to basic operations of the computer simulation. The objective of the simulation was to increase the individual's life expectancy by making changes in the individual's health habits and lifestyle.

During the fourteen day treatment period students were taught for half an hour using structured teaching in a conventional instructional setting. A modified version of the Healthways Student Workbook (Carnine, Wong, & Carnine, 1983) was used. The intent of instruction within the classroom setting was to offer a deeper understanding of health concepts, vocabulary, and basic correlations between health habits and diseases.

Next students spent a half hour in the computer lab. Students in the treatment group spent five to ten minutes learning an explicit problem solving strategy to be used while playing the Healthways computer simulation. Then they moved to the computers to work on each of the simulations. One computer was available for each student.

The control group received the same instruction in the classroom and also received the same set of simulations in the computer lab. However, instead of learning an explicit problem solving strategy, they reviewed the basic vocabulary from the computer simulation.

During the course of instruction, students were administered six test health profiles to determine performance on a breadth of health simulations. Following fourteen days of instruction students were administered two different tests. As well students were individually interviewed to determine their attitudes toward this form of instruction.

RESULTS

The Nutrition and Disease test was administered as both a pretest and as a posttest. This test focused on gains in content knowledge that occurred through the course of instruction. A 2 X 2 analysis of variance (ANOVA) with repeated measures was performed on the total number of correct responses on the pretest and posttest. This statistic was utilized to determine if learning did occur over the time period between these tests and if there was a significant difference between groups in learning basic health facts. No significant difference between groups was hypothesized as both groups received the same core health curriculum.

The results of the 2 X 2 ANOVA with repeated measures indicate a significant effect ($p < .001$) for the repeated measure, suggesting that both control and experimental groups improved significantly over the instructional period. No significant difference was found between groups on this measure. The learning gains made by both groups are summarized in Table 1.

TABLE 1. Means (M) and Standard Deviations (SD) for Each Group on the Nutrition and Disease Pretest and Posttest

Group	N	Pretest M†	Pretest SD	Posttest M†	Posttest SD
Experimental	10	10.6	6.4	18.2	6.3
Control	11	11.1	6.5	18.0	6.0

†Maximum possible score = 30

Students played a series of six Healthways simulation tests through the course of the study, approximately one every three days. A significant difference ($p < .05$) was found between groups on the percentage of wins on the simulation tests. The experimental group won approximately 61% of the games played (Mean = 61.20, SD = 26.11), while the control group won approximately 44% of the games played (Mean = 44.00, SD = 21.28). These results suggest that students given an explicit problem solving strategy were better able to perform on the computer simulations.

Upon completion of two weeks of instruction, students were given a test to determine their ability to generalize problem solving skills from the computer simulation. Students were presented with two different videotaped scenarios of individuals having informal discussions that, through the course of conversation, conveyed an individual's health profile. Students watched each videotape twice, then wrote three recommended health changes, in a prioritized order, and a reason for making each of these changes.

A one-tailed t-test performed on this measure did not yield a significant difference between groups. Although the means showed signs of an existing difference (Experimental Mean = 17.60, Control Mean = 12.82), the variance within groups (Experimental S.D. = 8.30, Control SD = 9.42) was sufficient to yield a nonsignificant statistical result. The results of this measure are clouded by a lack of statistical power, due to a low n in the study. More research is needed to clarify this result.

Some interesting results were obtained by analyzing the specific scoring criteria within the measure. Students in the experimental group performed at a significantly higher level ($t = 2.83$, $p < .01$) than the control group on identifying needed health changes. Although neither group was able to prioritize required health changes effectively, the experimental group was better able to identify appropriate changes.

Upon completion of the study, students were given a structured, individual interview. Data from this interview suggest a consistently positive attitude toward the health program and particularly the computer simulation. Eighty-two percent of students indicated they enjoyed studying using the Healthways computer simulations. Ninety-one percent indicated they would be interested in studying more about health using the computer. In contrast, only eighteen percent wanted to study more health if they could not use the computer. This level of interest was also evidenced during the study as students requested the simulation outside of class time. This interest level is not interpreted as a novelty effect. The majority of students had used computers on a regular basis in school for over a year.

IMPLICATIONS OF THE STUDY

Three implications of this study are briefly examined: (a) the effect on student attitude, (b) the effect on student learning, and (c) the teacher's role when using a computer simulation.

Effect on Student Attitude. The data from this study parallel findings of other studies that also examined student attitude. Hollingsworth (1988) found similar results on a questionnaire following a pilot study of geography materials that used hyper-media software. All students (n= 8) indicated they would be interested in studying geography if they were to continue using the computer. Only 25% reported an interest in continuing to study geography if they could not use the computer.

Seymour, Sullivan, Story and Mosley (1987) found similar results in an experimental study comparing attitudes of students using pencil/paper versus students using computers in a science drill and practice exercise. All factors were constant between the two groups, except for presentation by computer or by pencil/paper. The authors found that students in the computer group rated learning as more interesting than students in the pencil/paper group. As well, nearly all students indicated a preference to do subsequent exercises using the computer rather than using pencil/paper.

Motivation is central to learning. This is particularly relevant within special education settings where many students have typically faced repeated failures, resulting in low levels of motivation. The results of the current study support other findings, suggesting that the use of appropriate instructional software is more motivating than traditional paper/pencil forms of instruction.

Effect on Student Learning. Student learning may be examined as two separate factors: learning of content knowledge and learning of procedural knowledge. Results of the Nutrition and Disease Test indicate learning gains in the area of content knowledge. Students demonstrated improved understanding of health facts and concepts on this test. These findings are supported by Woodward, Carnine and Gersten's work (1988) which used the same test with senior high school students.

The effect of this instructional approach on student learning of procedural knowledge is unclear. Students in the experimental group had acquired the decision making skills taught in the classroom. This was demonstrated by their performance on the six Healthways simulation tests. However, they were unable to effectively generalize this information into the novel environment of videotaped profiles. Although they were able to identify the health habits that required change for one profile, they did not demonstrate the more advanced cognitive skill of prioritizing health changes.

The Teacher's Role. Tom Snyder (1988) underlines the importance of the teacher's role in utilizing computers in the classroom. He suggests that instructional software designers may be developing experiences that ignore the teacher. Teachers do offer an important and central role in the classroom. They need to have influence in the instructional process and act as a major contributor to this process.

Researchers have found that instruction using computer simulations alone, simulations that are not part of an integrated curricular approach, have not yielded improved learning over conventional forms of instruction (Woodward, 1985). The instructional power of computer simulations may reside in the effective combination of conventional instruction and computer instruction. Computer simulations may provide the opportunity to

solidify and enhance content knowledge through the development and practice of procedural knowledge. To achieve this combination we must recognize the importance of integrating computers into the curriculum, rather than attempting to supplant conventional instruction with computer instruction.

REFERENCES

1. Carnine, D., Lang, D., & Wong, L. (1983). Health Ways. Unpublished computer software. University of Oregon, Eugene.

2. Carnine, L., Wong, L., & Carnine, D. (1983). Health Ways Student Workbook. Unpublished manuscript. University of Oregon, Eugene.

3. Carnine, L., Wong, L., & Carnine, D. (1984). Health Ways Teacher's Guide. Unpublished manuscript. University of Oregon, Eugene.

4. Clark, R.E. (1985). Confounding in educational computing research. Journal of Educational Computing Research, 1(2), 137-148.

5. Clark, R.E., & Voogel, A. (1985). Transfer of training principles for instructional design. Educational Communication and Technology: A Journal of Theory, Research and Development, 33 (2), 113-123.

6. Coombs, H. (1986). Computer Assisted Health Counselling: A New Opportunity. Paper presented at Fifth Canadian Symposium on Instructional Technology, Ottawa, Canada.

7. Engelmann, S. & Carnine, D. (1982). Theory of Instruction: Principles and Applications. New York: Irvington.

8. Jolly, B. (1982). A review of issues in live patient simulation. Programmed Learning & Educational Technology, 19, 99-107.

9. Hollingsworth, M. (1988). Hyper-media field test results. Unpublished pilot study. (Available from author: Educational Psychology, University of Calgary, AB)

10. Horner, R., Jones, D., & Williams, J. (1985). A functional approach to teaching generalized street crossing. The Journal of The Association for Persons with Severe Handicaps, 10(2), 71-78.

11. Norman, G. R., Muzzin, L. J., Williams, R. G., & Swanson, D. B. (1985). Simulations in health sciences education. Journal of Instructional Development, 8, 11-17.

12. Roblyer, M.D. (1985). The greening of educational computing: A proposal for a more research-based approach to computers in instruction. Educational Technology, 25(1), 40-44.

13. Salomon, G. (1984). Computers in education: setting a research agenda. Educational Technology, 24(10), 7-11.

14. Seymour, S.L., Sullivan, H.J., Story, N.O., & Mosley, M.L. (1987). Microcomputers and continuing motivation. Educational Communication and Technology Journal, 35(1), 18-23.

15. Snyder, T. (1988). Tools for Teachers. The Computing Teacher, 16,(1), 8-16.

16. Stokes, T., & Baer, D. (1977). An implicit technology of generalization. Journal of Applied Behavior Analysis, 10(2), 349-367.

17. Woodward, J., Carnine, D., & Gersten, R. (1988). Teaching problem solving skills through computer simulations. American Education Research Journal, 25(1), 72-86.

18. Woodward, J P. (1985). Teaching health concepts and problem solving skills through effective instructional practices and a computer simulation. Unpublished doctoral dissertation; University of Oregon, Eugene.

INTEGRATING TECHNOLOGY IN THE SOCIAL STUDIES CURRICULUM: THE EFFECTS OF USING MULTIPLE APPLICATIONS TO ENHANCE LEARNING

G. Witwicki, L. Katz
Computerized Sport Systems Group
The University of Calgary
Calgary, AB

Emphasis in computer education has advanced from one of literacy to integration of the computer into the school curriculum--an exciting and potentially powerful shift. However, a careful study is necessary to examine the most efficient and effective way to implement this new philosophy.

This research study used simulation and database applications with elementary school aged children in a social studies environment. Students were taught to organize and classify world geography material from the elementary school social studies curriculum. The study examined simulations as a means of providing an interesting context through which students could explore the subject, subsequently using a database to organize and utilize information obtained from the simulation. The treatment group used a computer simulation to provide a context for the introduction of databases, while the control group used a traditional approach. Pre-test and post-test measures of computer attitude, social studies attitude, computer experience, geography and organizational skills were obtained.

Results suggest that providing a context for collection, classification and organization of information creates a motivating and stimulating environment leading to enhancement of skills as well as acquisition and retention of knowledge.

L'INTÉGRATION DE LA TECHNOLOGIE DANS LE PROGRAMME DE SCIENCES HUMAINES: DIFFÉRENTES APPLICATIONS POUR PARVENIR À UN MEILLEUR APPRENTISSAGE; LEURS EFFETS

Le rôle de l'informatique dans l'enseignement a aujourd'hui dépassé le stade de la simple sensibilisation à l'ordinateur pour se porter sur son intégration dans le programme même des études scolaires - un tournant passionnant et riche de possibilités. Il importe toutefois de mener une enquête sérieuse pour savoir quels sont les moyens les plus efficaces qui permettront d'assurer cette nouvelle orientation.

La présente enquête a mis à profit l'application de simulation et celle de bases de données au niveau d'une classe élémentaire pour en programme de sciences humaines. Les élèves avaient appris à organiser et à classer certaines leçons de la géographie du monde tirées de leur programme d'études. Les simulations avaient pour but de leur fournir un contexte intéressant qui les mettait à même d'approfondir le sujet. Ils devaient, par la suite, se servir d'une base de donnés pour organiser et utiliser les informations obtenues à partir de la situation. Le groupe expérimental s'est servi d'une simulation informatisée comme contexte donnant lieu à l'introduction des bases de données, tandis que le groupe de contrôle a suivi l'approche traditionnelle. Ont été évaluées avant et après l'expérience les attitudes face à l'ordinateur, les attitude envers les sciences humaines, les expériences informatiques, les connaissances géographiques et les qualités d'organisation.

Les résultats permettent de conclure que la création d'un contexte prêtant au recueil, à la classification et à l'organisation des faits crée un environnement stimulant menant à un apprentissage plus riche ainsi qu'à une acquisition accrue et plus durable de la connaissance.

INTEGRATING TECHNOLOGY IN THE SOCIAL STUDIES CURRICULUM: THE EFFECTS OF USING MULTIPLE APPLICATIONS TO ENHANCE LEARNING

G. Witwicki and L. Katz

The University of Calgary

Computers can affect children in a variety of ways including increasing their motivation, attention span, participation, enthusiasm, and self esteem. In order to achieve their educational objectives, computer programs must effectively engage children in the instructional task (Norton and Resta, 1986; and Dunne, 1984). The basic value of technology lies in the way in which it is integrated into the classroom. While there is substantial anecdotal support for the classroom use of computer applications such as database management and simulation, the advancement of educational goals can only be accomplished with well developed strategies and tactics.

A database is an organized body of information (Caissy, 1988). According to Lengel (1987) a database does not teach thinking, but a good teacher, armed with databases is in a better position to teach thinking than one without these resources. Budin, Taylor and Kendall (1987) contend that the use of databases enables students to gain the ability to organize data sequentially, test hypotheses, and look for trends. Eiser and Salpeter (1988) suggest that databases focus on the ability to draw conclusions about relationships.

Computer simulations are structured imitations of selected reality (Gredler, 1986). The use of computer simulations serve as a tool for teaching thinking and problem solving, reference skills, and, as a method of involving students in interactive programs. Norton and Resta (1986) believe that simulations in which concepts are embedded, go beyond games to create learning environments which are engaging, entertaining and interactive. Simulations in the classroom automatically fulfill one of the students' greatest desires, that school or learning be "fun", thereby encouraging students to learn. (Garfield, 1987).

Massialas and Papagiannis (1987), supported by Resnik (1984), advocate that a social studies curriculum which emphasizes process, represents the intersection of skills and knowledge drawn from the social sciences and humanities. In this context, the computer can be used to assist students to learn higher level thinking skills. A well designed program can engage students in collecting and interpreting data, debating and making collaborative decisions, and developing critical thinking skills which are central to social studies (Lengel, 1987; Dockterman and Snyder, 1986). When students use computers to look at historical events or social forces, their learning becomes more active, they retain more information for longer periods, and they develop better thinking skills (Solomon, 1986). Geography software, for example, offers students the opportunity to travel via computer and at the same time it reinforces their concept and knowledge of the world.

This research study examined the use of a world geography simulation to provide a context for the introduction and teaching of database applications with elementary school aged children. Geographic information obtained in the simulation was used in a computer database. The educational objective was to teach the concepts of organization, management, and classification within a social studies environment.

Methodology

The study was conducted in an elementary school with two separate Grade 5 classes in a lab consisting of Apple IIe computers. The treatment group consisted of 13 boys and 14 girls; the control group had 14 boys and 14 girls, mean age was 10 years. The treatment group was given a computer simulation, "Where in the World is Carmen Sandiego?", produced by Broderbund. The treatment group used the information obtained from the simulation in subsequent lessons on the use of databases. The control group was given traditional instruction on geography, and then used the information from the geography classes in the ensuing lessons on database management.

The Broderbund simulation comes with the World Almanac and Book of Facts, a world map, and a Police Dossier providing information regarding the 10 suspects encountered. The books serve as a comprehensive reference guide to assist students in relating clues to cities and countries on the map. Students take the role of detective, utilizing existing information and resources, and applying research skills to identify and locate the perpetrator of the identified crime. In order to maximize the use of research materials, students were organized into groups of three. Successful completion of the simulation required the students to work co-operatively. The control group was informed that they would be able to use the simulation at the conclusion of the study.

Both groups were given the same unit lesson on databases. The duration of the teaching unit was one month, which included one one-hour lesson on the use of the simulation, and three one-hour lessons on the database unit. Students in the treatment group had access to simulation during their free time, for one week prior to the database lessons.

Four evaluation instruments were used: (1) a computer attitude questionnaire; (2) a social studies attitude questionnaire; (3) a computer experience inventory; and, (4) a comprehensive test which measured geography and organizational skills. These four measures were obtained from each student pre- and post-intervention.

The hypothesis was that the use of simulations would provide a context for learning organizational skills in a subsequent unit on database management. Thus, students in the treatment group should show more improvement than students in the control group, with regard to both their knowledge of geography and their ability to gather and organize information.

Results

Computer Attitude Questionnaire

Children in both groups had a high positive attitude to computers in the pre-test, with the treatment group showing a significantly higher pre-test attitude ($F=5.12$; $df=1,51$; $p<.05$). However, both groups still showed a significant increase in attitude to computers from pre- to post- ($F=6.72$; $df=1,51$; $p<.05$).

Comprehension Test of Geography Knowledge and Organizational Skills

The treatment group had a significantly higher pre-test score on the comprehensive test ($F=5.70$; $df=1,51$; $p<.05$). Even though the treatment group had a higher pre-test score than the control group, both groups showed a significant increase pre- to post- on the comprehension test ($F=94.22$; $df=1,51$; $p<.001$). Moreover, the treatment by pre-/post-interaction on the comprehensive test was significant ($F=8.59$; $df=1,51$; $p<.01$) with the treatment group showing a much larger increase than the control group. This finding supports the major hypothesis.

Other Findings

Interestingly, the control group showed a significant drop on post-test evaluation of computer experience ($F=5.05$; $df=1,51$; $p<.05$). There were no significant differences between boys and girls on any of the measures.

Discussion

The results suggest that the use of simulations and database management programs are indeed powerful tools which can be used independently and effectively at the Grade 5 level. The study also supports the hypothesis that computer tools can be linked together to enhance learning. Children have generally positive attitudes toward computers and these attitudes tend to increase when children are exposed to useful applications in a

meaningful context. The control group used the database management tool and showed significant improvement in organizational and data management skills. However, the treatment group, which used the combination of computer simulation and database management, demonstrated an even larger increase in comprehension and organizational ability. Thus it would appear that computer simulations and database management can be used together to teach organizational and management skills within a social studies context.

The study was completed in Fall of 1988. Follow-up observations have shown that students continue to come in before and after school to gain access to the computers in order to play the simulation. Moreover, children from Grades 2 through 6 have started to play the simulation on a regular basis, and a Carmen Sandiego Club has been formed in order to buy new versions of the game.

To survive and thrive in an information based society will require that students learn skills and techniques for handling information. Computer programs such as database management tools and simulation games can be used to enhance the learning experience, and to teach children to manage information and to develop problem solving skills.

REFERENCES

Budin, Howard; Taylor, Robert; Kendall, Dianne. (1987). Computers and Social Studies: Trends and Directions. The Social Studies January/February 1987.

Caissy, Gail A. (1988) Using Databases in the Classroom. Computers in Education June 1987.

Dockterman, David; & Snyder, Tom (1986) How to Incorporate the Computer into the Social Studies Classroom. ERIC Document Number ED 271 372.

Dunne, James J. (1984) Gaming Approaches in Educational Software: An Analysis of Their Uses and Effectiveness. ERIC Document Number ED 253 207.

Eiser, Leslie & Salpeter, Judy. (1988) Where on Earth is Moncton, NB? Computers In Education March 1988.

Garfield, Robert (1987) Using Simulations in the Classroom: Teaching World Civilizations Through the Machine. History of Microcomputer Review.

Gredler, Margaret Bell. (1986). Computer Simulations. Educational Technology April 1986.

Lengel, James G. (1987). Thinking Skills, Social Studies and Computers. The Social Studies January/February 1987.

Massialias, Byron G. & Papagiannis, George J. (1987) Toward a Critical Review of Computers in Education: Implications for Social Studies. The Social Studies January/February 1987.

Norton, Priscilla & Resta, Virginia. (1986) Investigating the Impact of Computer Instruction on Elementary Students' Reading Achievement. Educational Technology March 1986.

Resnik, Hank (1984). From Social Studies to Social Sciences. Learning Oct. 1984.

Solomon, Gwen (1986) Playing with History. Electronic Learning. May/June 1986.

MICROMEC: UN ENVIRONNEMENT INTELLIGENT À BASE D'OBJETS POUR L'ENSEIGNEMENT DE LA MÉCANIQUE NEWTONIENNE

C. Coulombe, A. Coulombe, M. Décary
P. Desautels, R. Desautels, R. Young

APO Québec, Le Collège de Rosemont,
Machina Sapiens Inc.
Montréal, PQ

MicroMec est un simulateur graphique interactif destiné à l'enseignement de la mécanique newtonienne aux niveaux secondaire et collégial. Il intègre une banque de problèmes et un simulateur graphique agissant sur un micromonde de la mécanique newtonienne. Dans beaucoup d'applications pédagogiques, particulièrement pour l'enseignement des sciences, il peut être très intéressant de disposer d'un puissant outil de simulation qui permette aussi des manipulations directes par l'étudiant.

Le langage orienté objet ZOO II (pour Z Orienté Objet) a été spécialement développé pour implanter MicroMec sur le micro-ordinateur Macintosh d'Apple. Ce langage offre des outils de traitement mathématique dont un éditeur de formules mathématiques, un mécanisme automatique de propagation des erreurs de mesure (notions de chiffres significatifs et de calcul d'erreur) et la conversion automatique des diverses unités et grandeurs physiques.

Ce logiciel est le fruit d'une collaboration entre la compagnie Machine Sapiens Inc. (programmation) et le Collège de Rosemont (contenu, pédagogie). La Fondation Apple a prêtée du matériel informatique. Enfin, cette recherche a été financée et supervisée par le Centre Québécois de Recherche en Applications Pédagogiques de l'Ordinateur (APO Québec).

MICROMEC: AN INTELLIGENT OBJECT BASED ENVIRONMENT TO TEACH NEWTON'S MECHANIC

MicroMec is a graphic interactive simulator to teach Newton's mechanic at college level. This software is made up of a problems database and an interactive graphic simulator operating on physics microworlds. In many teaching tasks, particularly in sciences, it would be very useful to have powerful simulation tools that allow direct manipulation by students.

A new object oriented language, called ZOO II (for Z Orienté Objet) was developed for implementating MicroMec on the Apple Macintosh microcomputer. This language offers mathematics oriented tools such as a formula editor, an automatic mechanism for errors propagation (error calculus) and the automatic conversion of units and measure.

This work was realized by a team from Machina Sapiens Inc. (programmation) and by Collège de Rosemont (subject, pedagogy). Some computer equipment was loaned by the Apple Foundation. The work was supported by a grant from the Centre Québécois de Recherche en Applications Pédagogiques de l'Ordinateur (APO Québec).

MICROMEC: UN ENVIRONNEMENT INTELLIGENT À BASE D'OBJETS POUR L'ENSEIGNEMENT DE LA MÉCANIQUE NEWTONIENNE

C. Coulombe, A. Coulombe, M. Décary
P. Desautels, R. Desautels, R. Young

APO Québec, Le Collège de Rosemont,
Machina Sapiens Inc.
Montréal, PQ

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

ACTIONMARK: A KNOWLEDGE BASED PROGRAM FOR TEACHING AND COACHING SPORT AND PHYSICAL ACTIVITY

J.N. Vickers, S. Abbott, M. Morrill
Neuro-motor Psychology Laboratory
University of Calgary
Calgary, AB

ActionMark is a physical skills acquisition system developed at the University of Calgary. This rare application of software to the physical education setting actually places a computer in the learning environment, allowing the athlete to individually view his/her sequence of objectives and indicate technical and performance progress to the system. The exemplary use of the Apple Macintosh user interface allows course designer, instructor, and athlete to work with the databases of objectives and class performance through the same software. Presentation will contain a brief development overview, demonstration of the innovative user interface, and findings from use at the University of Calgary with university level physical education courses.

ACTIONMARK: UN PROGRAMME À BASE DE CONNAISSANCES POUR L'ENSEIGNEMENT D'ACTIVITÉS PHYSIQUES ET L'ENTRAÎNEMENT SPORTIF

ActionMark est un système d'acquisition d'habiletés physiques mis au point à l'Université de Calgary. Cette application exceptionnelle d'un logiciel à l'éducation physique introduit un ordinateur en milieu d'apprentissage et permet à chaque athlète de visualiser sa propre série d'objectifs et de signaler au système son évolution, tant au niveau de la technique qu'à celui de la performance. Cette utilisation remarquable d'un interface utilisateur Apple Macintosh permet au concepteur de cours, à l'instructeur et à l'athlète d'utiliser les bases de données d'objectifs et de performance de classe par l'intermédiaire du même logiciel. L'exposé comporte un bref aperçu des travaux de développement, une démonstration du nouvel interface utilisateur et une présentation des conclusions découlant de l'utilisation dans le cadre des cours d'éducation physique de niveau universitaire, à l'Université de Calgary.

ActionMark: A knowledge based program for teaching and coaching sport and physical activity

Joan N. Vickers, Shawn Abbott, Mark Morrill

Neuro-Motor Psychology Laboratory
University of Calgary

ActionMark is the first in a series of software programs designed to assist the teacher or coach in the school, club and university setting, by using the computer as a vehicle for showing what is to be taught and as a means of facilitating evaluation and feedback. The design of ActionMark is derived from the eight modules of an instructional design model called the *KS Model* (Vickers, in press) shown in Figure 1.

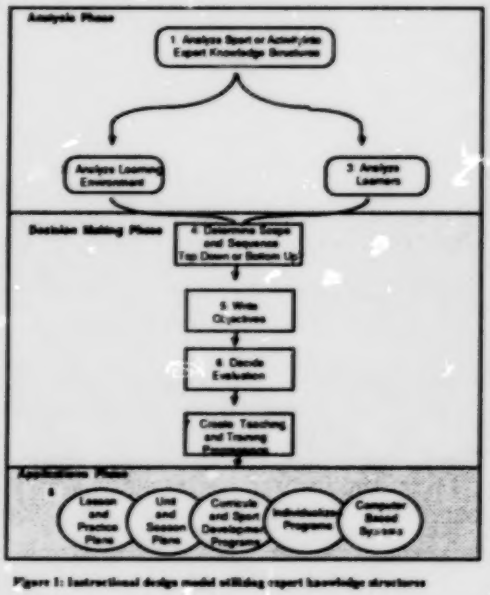


Figure 1: Instructional design model utilizing expert knowledge structures

ActionMark was developed on the Macintosh by Shawn Abbott and Mark Morrill using MPW object oriented Pascal. The development of ActionMark was completed by breaking it down into modules. "AppShell", written by Mark Morrill, (similar to MacApp), provided the overall structure of the application. Database functions were provided by Inside/Out (Shana Corporation). The user interface is managed by "UDialogue", a software unit developed by Shawn Abbott (Abbott, 1989). UDialogue is a prototyping and development tool for high-quality Dialogues. A Dialogue may be modified *while the software is running*, both in appearance and function. At any time while a Dialogue is in use, the software developer may toggle between user mode and authoring mode for that Dialogue. This "runtime authoring" comes in the form of a direct manipulation graphical interface.

The functions of ActionMark are as follows:

1) ActionMark is an activity specific program that comes complete with a knowledge structure for each sport or activity.

2) Specific skills, strategies and concepts for each activity are derived from a hierarchical knowledge structure developed for each sport or activity.

3) ActionMark presents the knowledge structure in a list format, thus permitting the teacher or coach to scroll through the components and set a scope and sequence for a course, a unit, a training program, or to meet individual needs.

4) Each skill, strategy, and concept identified in the knowledge structure has associated with it a progression of objectives, listed novice to elite, with evaluation levels recommended for each objective.

5) The instructor may accept the objectives as presented or tailor their instruction by varying the weight of an objective and/or by setting performance standards that are more appropriate for their context and student/athletes.

6) The student interface has been designed to function in a gymnasium or other facility where instruction, practise and/or play is occurring. Only one machine is necessary for a class of 20 students.

7) Students input their own scores, as earned in the performance objectives or awarded by the instructor in the technical objectives.

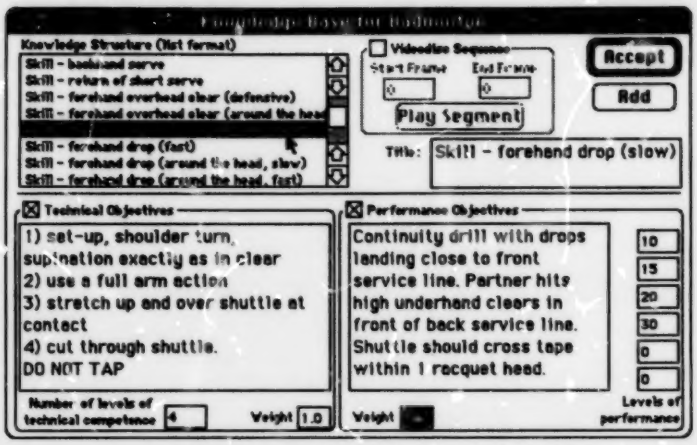


Figure 2. Setting up the knowledge base in ActionMark

8) Improvement in each skill, strategy or concept is facilitated through the use of a formative evaluation system. Improvement in each skill, strategy or concept is communicated to each student, as well as for the program overall.

9) ActionMark also comes with banks of questions permitting the setting of tests in rules, techniques, and/or strategies. These questions are also designed to be answered in the gymnasium environment.

10) ActionMark maintains a record of all scores and provides the necessary printouts for both students, teachers, and parents.

11) A videodisc interface is included with the program, thus making possible the showing of visual examples of selected skills, strategies and concepts (videodisc permitting).

Setting up the knowledge base for ActionMark

Figure 2 shows the screen used to input the knowledge base for ActionMark. Shown as an example is badminton (Vickers and Brecht, 1987). In the scroll box entitled *Knowledge Structure (List Format)* are listed the components of the knowledge structure for badminton. Any number of skills, strategies and concepts may be entered, with each input through the box entitled *Title*. Following this, both *Technical Objectives* and *Performance Objectives* are entered. The weight of each objective may be set to reflect the degree of difficulty of the skill, strategy or concept. In addition, the number of levels of competence or performance may also be set, in levels ranging from 1- 6. Finally, toward the top of the screen is the *Videodisc Sequence* box, where the appropriate start and end frame numbers may be entered for each objective.

Entering the names of students and designing a course, a unit, or individual training program.

The screen shown in Figure 3 has two functions. First, you are able to enter the names of your students and second, you are able to design a specific course, unit or training program for them. Each student's name, ID number, and other pertinent information is entered. Following this you may then design a specific course derived from the overall knowledge base. A course or unit or training program is created by selecting knowledge structure components from the *Design a Course* list (see figure 3) and tests from the *TestBank*.

This is accomplished by double clicking on the skills, strategies, concepts, and tests desired. After a skill, concept or strategy is selected (this is indicated by the bullet • symbol), then it is possible to activate or de-activate the objectives by clicking the *Use Technical* and *Use Performance Objective* boxes shown at the bottom of the screen. A histogram/bar graph appears in order to indicate student progress through that component. This screen also lets the instructor quickly see each student's progress on the course at any time. They may also leave messages and receive them from students.

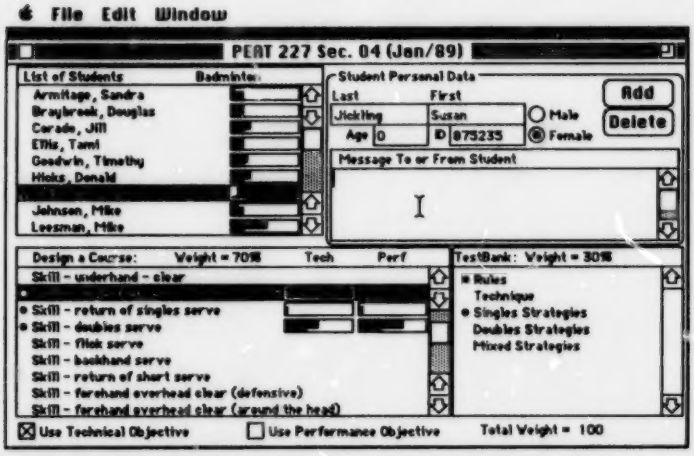


Figure 3. The ActionMark instructors interface for course design

The students screen.

Figure 4 shows one of the two screens the student sees. Shown here is their complete *Course of Study*. All knowledge components with technical and performance objectives are listed. Progress in each technical and performance requirement is shown in the histogram to the right. The grey bar of the histogram shows the average score of the class to date on each skill. The black bar shows the student's progress. At the bottom are shown the specific requirements for each technical and performance objective. The student's progress is recorded by clicking their level of achievement on the thermometer. At the top, is a statement summarizing what the student has attained to date, both as a raw score and percentage, and their progress relative to others in the class on set criterion or national norms.

Setting and taking a test.

As shown in figure 3, the instructors interface for course design , ActionMark permits the setting of tests in rules, technique, and strategies. In most physical education or activity settings, there are occasions when students are waiting,

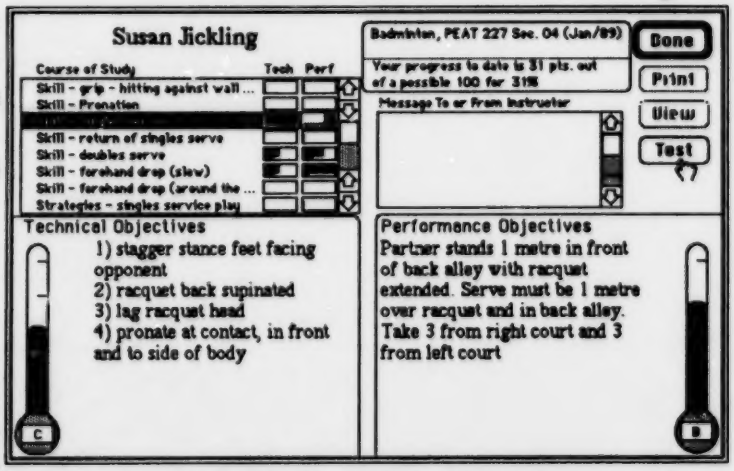


Figure 4. The ActionMark student's screen.

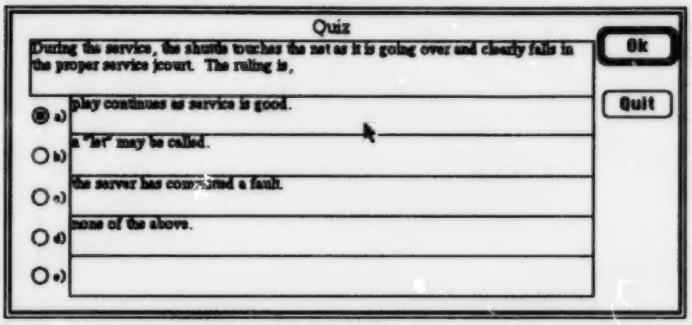


Figure 5. The test screen

resting or otherwise taking their turn to get on a court or be actively involved. ActionMark comes with banks of questions for rules, technique, and strategies. Before class begins, the instructor sets the parameters for a test, for example you may feel there is time for each student to be tested on five questions from the rules bank over the next three classes. Separate questions are then randomly generated for each student. The instructor can manipulate the number of questions generated per test, the number of times a test may be taken, the value of each, and the type of retests (for example removal of correctly answered questions from the bank), as shown in figure 6. Students receive their results on the test immediately and this score is automatically incorporated into their overall progress or formative report.

Summary.

ActionMark is a knowledge based program that comes complete with the information needed to design and implement instruction in a specific activity. ActionMark allows the instructor to design a unit or course of instruction to reflect skill level, time availability and number of students; it encourages the development of technical skill; it permits the incorporation of tests on knowledge of rules, technical skill and strategies, it creates a dialogue between students and instructors on the biomechanics of performance; it leads to an individualized teaching environment where students are soon involved in practising different skills and strategies according to need. ActionMark also comes with a videodisc driver allowing the showing of skills and learning activities instantly through this medium.

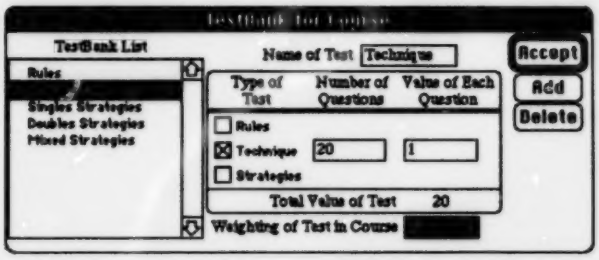


Figure 6: Setting the parameters for a test.

References

Abbott, Shawn. (1989). *Interim Report on UDialogue.* Paper presented at the First Pacific Universities Consortium Conference, Wollongong, Australia.

Vickers, J. & D. Brecht. (1987). *Badminton: a Structures of Knowledge Approach.* University of Calgary Printing Services, Calgary, Alta.

Vickers, J. (In Press). *Instructional Design for Teaching Physical Activity: A Knowledge Structures Approach.* Human Kinetics Publishers, Champaign, Ill.

LE DÉVELOPPEMENT D'OUTILS DE SIMULATION PÉDAGOGIQUE À L'AIDE D'UN PROGICIEL DE GESTION D'ENTREPRISE

C. Camirand, S. Carrier
La société d'éducatique Enr.
Mascouche, PQ

Cette communication présente les résultats d'une recherche ayant mené au développement de deux outils de simulation pédagogique à l'aide du progiciel "Excel". "Excel" possède des capacités de programmation permettant l'exécution automatique des fonctions du progiciel et cela constitue une programmation de type procédural. L'utilisation des cellules du tableur pour le stockage de l'information et la définition d'opérations simples constitue une programmation déclarative alors que l'utilisation de plusieurs fichiers simultanés constitue un environnement de programmation "orientée-objet".

Nous avons utilisé ces capacités pour réaliser deux simulateurs en enseignement des sciences pour les niveaux fin secondaire et collégial. Le premier est un simulateur de l'équilibre d'une réaction chimique et le second est un simulateur pour l'étude des images produites par des lentilles. Les deux systèmes permettent le contrôle de plusieurs variables et contiennent des outils de support à la recherche scientifique.

THE DEVELOPMENT OF EDUCATIONAL SIMULATION TOOLS USING A BUSINESS MANAGEMENT PACKAGE

This paper will present the results of research that led to the development of two educational simulation tools using the "Excel" package. "Excel" has programming capabilities allowing the automatic execution of the functions of the software package; this is programming of the procedural type. The use of the cells in the spreadsheet program for storing information and defining simple operations constitutes declarative programming, whereas the use of several simultaneous files is "object-oriented" programming.

We used these capabilities to assemble two science teaching simulators for the senior secondary and college levels. The first is a simulator of the equilibrium of a chemical reaction, while the second is a simulator for the study of images produced by lenses. Both systems allow control of several variables and incorporate tools useful in scientific research.

LE DÉVELOPPEMENT D'OUTILS DE SIMULATION PÉDAGOGIQUE À L'AIDE D'UN PROGICIEL DE GESTION D'ENTREPRISE

Charles Camirand, Serge Carrier

La Société d'Éducatique Enr.,
1115 Pincourt, Mascouche Qc, J7L 2I8

INTRODUCTION

Ce texte traite de l'architecture et de la conception de simulateurs théoriques interactifs utilisés pour l'enseignement de notions scientifiques à des étudiants de l'ordre secondaire et de l'ordre collégial. Il énonce les objectifs de la recherche et leurs raisons. Ce texte explique ensuite pourquoi ces simulateurs ont été réalisés à l'aide d'un progiciel de gestion d'entreprise et comment ce progiciel peut être perçu par rapport aux paradigmes courants de l'intelligence artificielle. Il laisse enfin quelques idées pour aller plus loin.

LES OBJECTIFS

Les objectifs visés dans cette expérience étaient les suivants:

-utiliser des simulateurs interactifs pour enseigner des concepts de science,

-créer et utiliser une boîte d'instruments informatisés et intégrés d'aide à la démarche scientifique,

-utiliser des progiciels pour prototyper rapidement ces simulateurs,

-faciliter le transfert d'informations entre l'environnement de recherche et l'environnement de communication écrite.

Il semble que l'utilisation de simulateurs interactifs pour la découverte de concepts scientifiques soit une démarche calquée sur la méthode scientifique, dans laquelle on accumule d'abord des données, puis on forme des concepts, où l'on établit ensuite des relations et des hypothèses, et enfin où l'on construit un modèle du phénomène. Le simulateur interactif permet de modifier une variable de façon répétitive sans obscurcir la connaissance et la démarche par des obstacles environnementaux tels l'appareillage et les variables non-identifiées du système. Cela permet à l'apprenant d'explorer un monde théorique et de recevoir rapidement une rétroaction. L'apprenant extrait du simulateur des résultats aussi rapidement qu'il peut les demander et peut former ainsi de façon concourante des concepts sur le phénomène. Le simulateur permet de tester facilement et rapidement des hypothèses (What if...?) dans le but de les confirmer ou de les infirmer.

Le simulateur pouvant fournir une très grande quantité de données volatiles en peu de temps, il est important de disposer d'un moyen de retenir celles qui sont intéressantes aux yeux de l'expérimentateur. De plus, la signification des données n'apparaît souvent qu'après un traitement plus ou moins long, comme une combinaison de séries de valeurs, la construction d'un graphique, etc. Il faut donc disposer d'outils capables de traiter rapidement ces données afin d'en extraire une image intelligible pour l'expérimentateur.

Les progiciels de type "tableur intégré" (Excel, Lotus-1-2-3, etc.) présentent un environnement de calcul rapide permettant une plus ou moins grande flexibilité dans la disposition de l'information. Leurs capacités de combinaisons mathématiques permettent de construire un modèle théorique caché du phénomène et d'en afficher certaines valeurs.

L'adjonction de macro-commandes augmente la puissance de ces progiciels jusqu'à les placer avec les langages de programmation dits de haut-niveau. De plus, les capacités graphiques de ces programmes accélèrent la fabrication d'outils de présentation des connaissances.

Recueillir, traiter et transporter l'information vers un milieu de communication (texte, acétate de présentation) constituent trois tâches souvent éloignées et irréconciliables. Il est alors rentable de disposer d'un simulateur interactif capable d'interchanger son information (textuelle et pictorale) avec un traitement de texte, un générateur d'animations, un traitement d'images, une base de données ou un analyseur statistique.

DEUX SIMULATEURS

En intelligence artificielle, les trois principaux paradigmes de travail sont la programmation fonctionnelle (et procédurale), la programmation déclarative et la programmation orientée-objet. La première se matérialise dans des langages comme LISP ou LOGO, la deuxième se retrouve dans Prolog et la dernière dans SmallTalk. Certains langages intègrent deux de ces paradigmes, comme LOOPS.

Un tableur comme Excel n'est pas un outil d'intelligence artificielle en soi, mais il peut être exploité de façon à exhiber certaines caractéristiques des langages mentionnées plus haut. Sa capacité à retenir des valeurs et à produire dans des cases des résultats calculés est de nature fonctionnelle, ses capacités de macro-commandes sont de nature procédurale, ses capacités d'extraction critériée d'une portion base de donnée touchent légèrement à la programmation logique et sa capacité à travailler sur différents documents contenant des structures potentiellement différentes est de nature objet.

Pour illustrer ces capacités, nous avons construit un document combinant ces approches. Il simule, de façon symbolique, une course entre différents véhicules, où chacun possède ses attributs propres et ses méthodes. Ils peuvent s'adresser et recevoir des messages et réagir en conséquence. La figure 1 illustre cette application d'Excel.

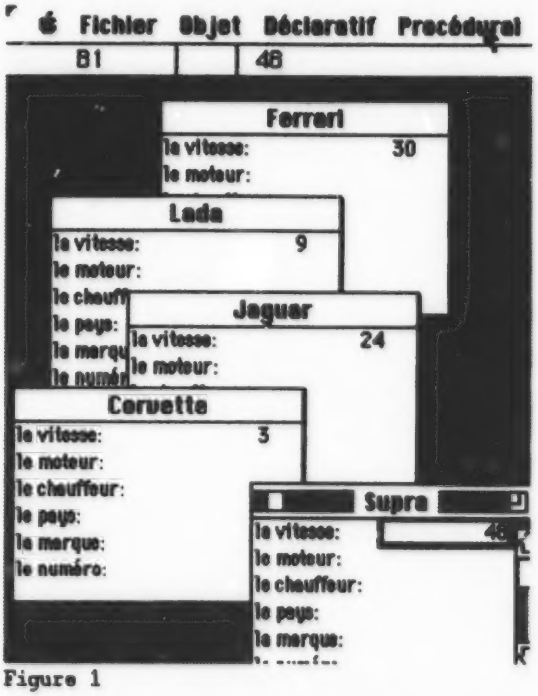


Figure 1

Miroirs

"Miroirs" est un document Excel en quatre "objets" permettant la simulation de l'interaction entre des rayons lumineux et un miroir courbé (figure2). Il dispose d'outils d'extraction et de traitement des connaissances, ainsi que de facilités d'exportation de ses données vers d'autres progiciels. Il permet de simuler la construction des images, d'analyser les caractéristiques d'un miroir et d'explorer la marche des rayons lumineux.

Équilibre chimique

"Équilibre chimique" regroupe aussi quatre "objets" dans un document Excel. Ces objets permettent d'explorer, dans un premier temps, par le biais d'une simulation théorique, le concept de l'équilibre chimique. Dans un deuxième temps, l'utilisateur peut découvrir

la notion de constante d'équilibre. Enfin, les deux derniers modules concernent le principe de Le Châtelier, c'est-à-dire l'effet sur l'état d'équilibre de modifications induites sur les conditions d'un système à l'état d'équilibre. "Équilibre Chimique" a été réalisé avec l'aide de APO-Québec.

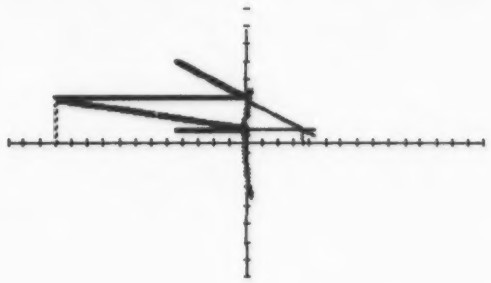


figure 2

CONCLUSION

Deux applications pédagogiques sérieuses sont sorties de cette expérience et le temps combiné de leur développement ainsi que de leur mise au point a été de beaucoup inférieur à ce qu'il aurait été, eussent-elles été réalisées par des méthodes conventionnelles. Pressenti originalement comme une simple feuille de propotypage, le document Excel est devenu un système viable et efficace. De plus, la perception d'Excel dans un contexte d'intelligence artificielle ouvre des façons nouvelles d'appliquer Excel dans le domaine de l'enseignement. Il est à espérer que d'autres chercheurs utiliseront ce type de logiciel ou d'autres types dans la même optique pédagogique.

BIBLIOGRAPHIE

Bergeron, A., et Paquette, G., Discovery Environments and Intelligent Learning Tools, in Artificial Intelligence and Education, Ablex, 1989.

Bracke, D., Utilisation du tableur pour l'étude du mouvement des projectiles, Le Bus, v.6 no.1, septembre 1988, p.11.

Camirand, C., Les utilitaires de demain, Le Bus, v.2 no.1, octobre 1984

Camirand, C., et Carrier, S., Deux unités d'apprentissage en physique et en chimie, Le Bus, v.6 no.1, novembre 1988, Les actes du colloque 1988, p.26

Camirand, C., Carrier, S., et Grosariu, N., ISI enseigné à l'aide des progiciels, Le Bus, v.6 no.1, novembre 1988, Les actes du colloque 1988, p.40

Carrier, S., La pomme chirale, Le Bus, v.2 no.3, février 1985

Le développement de dix unités d'apprentissage à l'aide de progiciels, rapport-final du projet LOUPE Phase 1., Micro-Intel et APO-Québec, Montréal, Décembre 1988

Devis de Louti: Un environnement pour la conception de logiciels-outils favorisant l'apprentissage par le traitement des connaissances, Micro-Intel et APO-Québec, Montréal, Août 1987.

Gomel, J., Le chiffrier et les statistiques, Le Bus, v.6 no.1, septembre 1988, p.21.

Paquette, G., Des progiciels d'applications aux systèmes à base de connaissance, Le Bus, v.6 no.1, novembre 1988, Les actes du colloque 1988, p.172

Tremblay, D., Le chiffrier en classe d'histoire, Le Bus, v.5 no.5, septembre 1988, p.6

TRAINING/FORMATION

DESIGN DECISIONS FOR CBT IN AN AIRCREW TRAINING SYSTEM

M.L. Ross
Link Flight Simulation Division
CAE-Link Corporation
Binghamton, NY, USA

The CBT system in an Aircrew Training System (ATS) is used to develop, deliver and manage portions of the instructional courseware. The design of the CBT system depends on requirements from two sources: ATS or government requirements for training (nonderived) and front-end analyses and equipment constraints (derived). All decisions as to the design of the CBT system are balanced between these requirements, sound instructional principles and human factors considerations.

The functional flow, or manner in which students are routed through lessons, controls the students progress through the courseware and, therefore, is an important factor in the design of the CBT system. The CBT system must interface with the various other functions of the ATS in order to develop and deliver the courseware and to collect the appropriate data for management and operation of the training. Physical design of the CBT system depends upon the specific application for the ATS in which it will be used. Another factor to be considered is the ability of the system to adapt to ATS growth and to emerging technology. Finally, cost and existing facilities must be considered in the choice of the system.

DÉCISIONS DE CONCEPTION VISANT L'EAO À L'INTÉRIEUR DU AIRCREW TRAINING SYSTEM

Dans le cadre du Aircrew Training System (ATS), le système EAO sert à concevoir, à produire, et à gérer des parties du didacticiel. La conception du système d'EAO dépend des exigences des deux parties : les exigences du système ATS ou du gouvernement en matière de formation (exigences générales) et les analyses de pointe et les restrictions d'équipement (exigences particulières). Toutes les décisions relatives à la conception du système EAO sont prises en fonction de ces exigences, de principes de formation solides et de facteurs humains.

Le déroulement fonctionnel, ou la manière dont les étudiants sont dirigés dans les leçons, permet de contrôler leurs progrès à l'intérieur du didacticiel et constitue de ce fait un élément important pour la conception du système EAO. Ce dernier doit être en relation avec les autres fonctions du système ATS pour qu'il soit possible de créer et de livrer un didacticiel et de recueillir les données appropriées à la gestion et à l'exploitation de la formation. La conception physique du système EAO dépend de l'application particulière du système ATS à laquelle il est destiné. Un autre facteur dont il faut tenir compte est la faculté du système à s'adapter à la croissance du système ATS et à la l'évolution constante de la technologie. En dernier lieu, il faut tenir compte du coût et des installations existantes au moment de choisir un système.

DESIGN DECISIONS FOR CBT IN AN AIRCREW TRAINING SYSTEM

Mary L. Ross

Operations and Training Analysis Group
Link Flight Simulation Division, CAE-Link Corporation
Binghamton, New York USA

INTRODUCTION

The CBT system in an Aircrew Training System (ATS) is used to develop, deliver, and manage portions of the instructional courseware. The design of the CBT system depends on requirements from two sources: ATS or user requirements for training (nonderived), and front-end analyses and equipment constraints (derived). All decisions as to the design of the CBT system are balanced between these requirements, sound instructional principles, and human factors considerations.

The functional flow, or manner in which students are routed through lessons, controls the students' progress through the courseware, and therefore, is an important factor in the design of the CBT system. Physical design of the CBT system depends upon the specific application for the ATS in which it will be used. The CBT system must interface with the various other functions of the ATS in order to develop and deliver the courseware and to collect the appropriate data for management and operation of the training. Another factor to be considered is the ability of the system to adapt to ATS growth and to emerging technology. Finally, cost and existing facilities must be considered in the choice of the system.

Recent improvements to computer hardware and software technologies have made it technically feasible and cost effective enough to allow their application in the development and delivery of ATS training. The use of computers in an ATS has several important advantages, including (a) standardization of delivery, (b) self-pacing, (c) greater student involvement in the instructional process, (d) objective performance testing, and (e) individualized remediation. Computers improve training effectiveness, make better use of available instructor resources, and increase the transportability of instruction from formal schools to field locations.

An ATS is a system designed and implemented by an independent contractor to provide training for military flight and maintenance personnel. In an ATS, computer-based training (CBT) refers to any training that is developed, delivered, and managed primarily by a computer. Because training of many tasks requires the students to view actual motion of real equipment, this paper considers interactive videodisc (IVD) as an integral part of CBT.

REQUIREMENTS

Requirements for a CBT system in an ATS come from two main sources: the user and the front-end analyses. The first set of requirements is firm (or nonderived). Nonderived requirements come primarily from the user through system requirement documents and statements of work. These requirements generally are high level and not sufficient for defining the complete system. The following are examples:

a. Maintain records of student performance and training

b. Generate student training reports

c. Major interfaces between system components shall be designed for maximum compatibility

The second set of requirements is derived from the various front-end analyses conducted in support of the user requirements. For example, if the analysis determines that accomplishment of many objectives requires the student to view motion sequences, then a requirement exists for a videodisc player, or similar equipment.

Some requirements are general ones that apply to all functions of the CBT system. These include such factors as security to limit user access; vendor support; color needs; and the need for operational linkage of two or more instructional stations.

CONTROL AND FUNCTIONAL FLOW

Control and functional flow refer to the methods in which students progress through the CBT program. Student interaction or input may be in the form of keyboard, touchscreen, or peripheral device such as a

joystick. The method used will be determined by the required application: paging/routing, part-task training, or evaluation.

For paging/routing, both touchscreen and keypad, a modified keyboard, should be available for the student. By itself, touchscreen becomes a tiresome method of progressing from display to display. Students, therefore, need the option of using a keypad or a touchscreen for selecting entries from the index and for paging or routing operations.

Part-task training includes activities requiring students to simulate training exercises such as activating switches. Touchscreen is the most effective method of input for these activities. In some activities, students must simulate movement. For example, students may need to simulate aircraft control input. This type of student input requires the use of a peripheral device such as a joystick.

For evaluation, all three types of input (touchscreen, keypad, joystick) may be used. Part-task performance evaluation requires touchscreen or peripheral device input. Standard test question answering requires keyboard/pad input.

DESIGN DECISIONS

The following paragraphs describe the configuration of two major CBT functions: authoring (development) station, and learning center (delivery) stations.

Authoring Stations. The authoring station will be used to develop, maintain, modify, and update the CBT courseware. Each authoring station includes standard CBT hardware and software and should be enclosed in a carrel to promote a quiet, efficient work environment. Specified stations should include additional hardware such as a scanner and videodisc player and monitor. These stations will be used for tryout of courseware during formative evaluation. All stations should interface with the host central processing unit (CPU) for accessing the support functions.

Learning Center Stations. A learning center (LC) is the academic hub of the ATS. It houses the CBT units and, generally, the student library as well as providing a quiet place to study. Both student stations and an instructor station will be required in the learning center.

The LC student station will be used for delivery and management of CBT lessons on line. To promote a quiet learning environment, each station should be enclosed in a carrel. Learning and student acceptability are enhanced by making the walls of the carrel a photographic representation of the aircraft cockpit. Each LC student station should interface with the host CPU for data exchange.

The instructor station (IS) will be used to monitor student progress/performance on CBT lessons and to manage delivery, evaluation, and maintenance data. The IS should be enclosed in a carrel to promote an effective work environment and to allow a measure of privacy for student-instructor consultations. Each IS should include all CBT hardware and software for a student station with the addition of a printer.

The IS hardware and software should interface with the host CPU for monitoring student performance, reviewing/printing evaluation data, receiving scheduling and maintenance data, sending/receiving mail, and receiving student requests for assistance.

CBT FUNCTIONAL INTERFACES

The CBT system must interface with all ATS training functions. Figure 1 depicts, on a high level, the manner in which the interfaces flow. Student stations and instructor stations must interface through a CPU with the IS and with the system support center for lesson delivery, for electronic mail, and to request assistance. The IS station also receives courseware updates, schedules, and maintenance data. The authoring station interfaces with the system support center for initial development and follow-on updates of courseware.

ADAPTION/GROWTH

An ATS life cycle may be as long as 15 or 20 years; therefore, the ATS CBT system must allow expansion/update to new inno-vative technology when it becomes avail-able. Artificial intelligence (AI) is one of the more promising technologies on the horizon. Because AI methodology and practices are not yet well-developed, this form of instructional delivery is not feasible for ATS use at this time; however, as it matures, it will be a desirable option.

FINAL DECISIONS

The analyst/designer writes specifications that outline the requirements for the proposed CBT system. These specifications will be used in procuring the CBT system. In any contract, cost is a major factor; therefore, after all requirements have been established, trade studies are conducted to identify systems that most closely meet the requirements. The results of the trade studies then are used in making cost-versus-requirement trade offs.

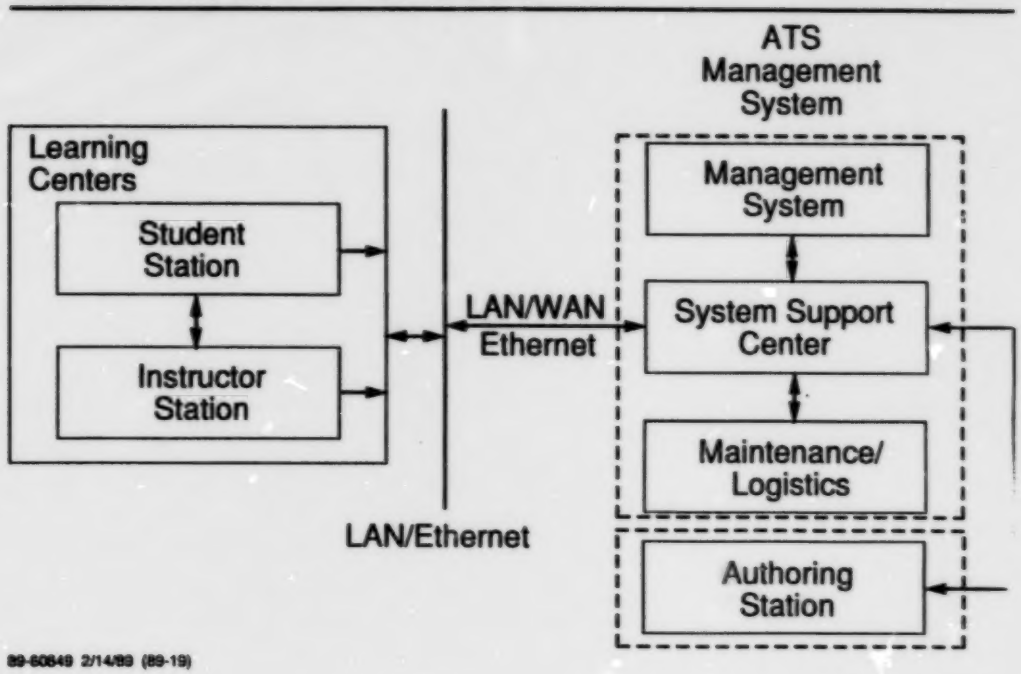


Figure 1. Functional interfaces.

SUMMARY

CBT is an important and valid media for training in an ATS. The system must be designed carefully to ensure that all the requirements for development, delivery, and management can be met in cost effective and efficient equipment.

REFERENCES

Carter, J., Ross, M.L., Smith, J., Guidelines for the Development of Computer-Based Training (Technical Document TD86-07). Seville Training Systems, Irving, Texas (1986).

Ross, M.L., CBT Concept Design Document (Technical Report C-17-00024). Link Flight Simulation Division, CAE-Link, Binghamton, NY (1988).

Ross, M.L., C-17 CBT Component Specification Document. Link Flight Simulation Division, CAE-Link, Binghamton, NY (1988).

PROJECT CTRAIN: CAL GOES TO SEA

R.J. Talbot
Paramax Electronics Inc.
Montreal, PQ

The Canadian Patrol Frigate Program is a multi billion dollar activity designed as the first part of a long term effort to equip the Canadian Navy for meeting the challenges of the next century. The operational environment is one of total systems integration. The paper will discuss our theoretical assumptions preceding courseware development and will outline how reality modified these assumptions. In particular, the following topics will be addressed: a) authoring system specification, development and modification; b) courseware development, evaluation and validation; 3) production of CAL for evolving systems; 4) the paper chase - development of documentation to support courseware; and 5) team roles in a high tech military project.

Approximately 100 lessons, each of one hour duration, will be used by all categories of ship's personnel.

PROJET CTRAIN: EAO S'EN VA-T-EN MER

Le Programme des frégates canadiennes de patrouille est une activité de plusieurs milliards de dollars et constitue la première étape d'un programme à long terme visant à permettre à la Marine canadienne de relever les défis du siècle prochain. L'environnement opérationnel en est un de complète intégration des systèmes. Le document traite des hypothèses sur lesquelles étaient basées l'élaboration du didacticiel et précise dans quelle mesure ces hypothèses ont été modifiées par la réalité. Les questions suivantes y sont traitées plus en détail : 1) spécification, mise au point et modification du système d'auteur; 2) mise au point, évaluation et validation du didacticiel; 3) élaboration d'EAO pour des systèmes évolutifs; 4) production de documents écrits - création de documentation à l'appui du didacticiel, et 5) rôles au sein des équipes dans un projet militaire à haute technologie.

Toutes les catégories de personnel des navires recevront environ 100 leçons de près d'une heure chacune.

PROJECT CTRAIN: CAL GOES TO SEA

R. J. TALBOT

PARAMAX ELECTRONICS INC.
MONTREAL, QUEBEC

PROJECT CTRAIN is the development of Computer Assisted Learning materials for the Canadian Patrol Frigate (CPF) Programme. The CPF programme is the largest Canadian naval ship-building programme in over a quarter of a century. It is also significant because, for the first time, the Canadian Navy has contracted out design and building, and crew training to the private sector. For the first time, too, CAL is playing a major role in training.

The Patrol Frigate is a highly sophisticated warship. A variety of sensors and weapons are co-ordinated through the Command and Control System (CCS). These sensors include radars (long and medium range, navigation, fire control), electronic support measures and sonar. Missiles (surface to air, surface to surface), 57 MM gun, torpedoes, electronic countermeasures, decoy systems and helicopters make up the weapons side.

The operator interface to the CCS is the Standard Display (Hardware) and the arrays and other software pages. The CTRAIN Project team was tasked to develop lessons for all personnel who will use the Standard Display or operator console. CTRAIN is the acronym for Console Operator Training and also refers to the custom designed authoring system.

METHODOLOGY:

A sailor loves a good story. If you are in the business of training sailors, you ought to be able to tell 'em well. Coleridge's ancient mariner could tell a compelling story:1

> It is an ancient Mariner
> And he stoppeth one of three
> "By the long grey beard and glittering eye,
> Now wherefore stopp'st thou me?
>
> The bridegroom's doors are opened wide,
> And I am next of kin;
> The guests are met, the feast is set:
> May'st hear the merry din."
>
> He holds him with his skinny hand,
> "There was a ship,"quoth he
> "Hold off! Unhand me, grey-beard loon!"
> Eftsoons his hand dropped he.
>
> He holds him with his glittering eye --
> The Wedding-Guest stood still,
> And listens like a three year's child:
> The Mariner hath his will.

The wedding guest is compelled to listen, and as the poem later recounts, to learn. If your trade is technical training, a story is the way to liven up the lesson. The miniscenario, as we call it, is a central feature of all our lessons. Instead of the Mariner's glittering eye we use glittering symbols on a screen. Retired, not ancient, storytelling mariners are part of the CTRAIN team. Most importantly, our stories are about ships to be like HMCS Halifax, Vancouver, and Calgary. We also stop, for training, about one in three!

CTRAIN lessons are intended to instruct, monitor and verify operator capabilities on the Standard Display. Each lesson consists of a series of frames linked, for the most part, linearly. The lessons are about "buttonology"; essentially, what the button does and when to use it. The Standard Display has what are known as fixed action buttons to perform standard functions such as changing range scales, hooking tracks and quick action buttons to select array functions. In our CTRAIN lessons we illustrate, sequentially, the use of the hundred or so arrays of the Command and Control System through simulation. The Standard Display is the delivery vehicle and the CTRAIN authoring system the development tool. To the student, a CTRAIN lesson is a short story that uses standard naval operations to illustrate control functions in a plausible and interesting manner. Or so we hope! As an added and necessary feature each lesson also contains elements of a tutorial dialogue with the lesson prompting the student for responses. Because of the highly procedural nature of Standard Display operation, we often ask for

recall of a previously demonstrated procedure in a different but analogous context. This is posed to the student as a problem:

> "You are the sensor weapons co-ordinator. Notice the hostile air track at 080,6DM. We need to determine engageability. Perform required actions."

If, in his answer sequence, the student makes an incorrect response, we cue or remediate to get him back on track. CTRAIN is designed to accept unique responses only.

At Paramax Electronics we have the perfect setting for our CTRAIN lessons. The Combat Systems Test and Support Facility (CSTSF) in Montreal contains life-sized physical and functional replicas of both the bridge and the operations room, the work environment. This is where the first operators will be trained. It is also where the entire system will be tested to ensure that man, machine and software will effectively meet threats at sea. Later, CTRAIN lessons will be used shipboard.

CTRAIN lesson development commenced in July 1987 with a preliminary scope based on operator requirements. This scope has evolved as our familiarity with the CCS increased and lesson boundaries were delineated. We use a modified instructional systems design (ISD) methodology. Our goal is to produce 70 lessons for use by 7 major categories of personnel by Spring 1989.

Our lesson development process is highly structured. We use a team approach in lesson design with expertize in Combat and Control System, naval operations, instructional design and evaluation. Because our lessons are technically complex, we prefer to develop and critique the entire lesson on formatted storyboards prior to data entry. Instructional design bugs are relatively easy to remove at this stage. We have identified 25 discrete steps for lesson completion. Formative evaluation includes both a technical assessment and a training materials critique from Paramax Standards. All Paramax instructional staff are encouraged to test lessons prior to student use. Our project group also includes two programmers and a data entry person.

CTRAIN AUTHORING SYSTEM:

CTRAIN is a custom designed authoring system. The authoring and presentation environments are separate. In the authoring of a CTRAIN, lesson data are entered into an hierarchically structured menu system. Lesson data and lesson functions are clearly separated and so lesson revision is easy. The authoring system, because of the console simulation requirement, accepts data to fit the display areas and there are, therefore, limits on text, numbers and graphics. CTRAIN has the full range of graphics, symbology and colour of the operational system. Lesson authors are not restricted by the system. In other words, lesson strategies i.e. examples, tests, summaries, etc. are totally under author control. Statistical analysis of quiz frame responses is automatic.

In the presentation environment, the lesson author has a number of debug options for lesson revision purposes. Loading a lesson is no more complex than the operation of a videocassette tape recorder.

IN RETROSPECT:

Our project is nearing completion. Course conduct will be in June/July. Thus no student performance data are available but formative evaluation checks using students have indicated that we are on track.

As lesson authors, we feel that we have learned and relearned some lessons ourselves. We base the following observations on the production of approximately 50 lessons or about 4,000 frames of software, 1,000 pages of support documentation and the scrutiny of 30,000 pages of computer printouts. Alternatively put, about 6 PYs have gone into the authoring effort. Our major observations fall into two major categories.

1. Instructional Design

   Develop Standards and Q/A Guidelines:

   Essential in a team effort. Our team operated out of Montreal and Minneapolis with added problems due to the transmission of confidential documents. We developed/collated detailed criteria for instructional design including: frame design, writing style and student performance checks.

   Experiment:

   We have rewritten Lesson No. 1 three times. Develop prototypes early. Our frame design techniques are still evolving as we address balance and proportion problems.

   Critique Page Turning Activities:

   Lessons must be interactive. When we proof our final storyboards we cast one member in the role of lazy student for an estimate of how much the student can avoid learning and

still complete the lesson. Other techniques include information/question ratio determinations.

Don't Forget Basics:

All lesson strategies involve some mixture of student attention grabbing, identification of lesson objectives, recall of prerequisite skills, subject matter presentation, learning guidance and performance checking. Questions should revolve around whether or not all these aspects have been addressed and the right balance made. This is basic for all lessons.

Simple ------> Complex:

Ordinarily, a trained Console operator will have on screen a plethora of symbology and data. In CTRAIN lessons we try to reduce clutter so that we can focus on particular symbology. Other techniques are word redundancy checks, KISS "lifeboat" checks, and graphics versus text options.

2. Project Management:

Team Composition:

Teams are formed because unique contributions are required from individuals. However, will likely get heated arguments in team efforts. Corrolary: need dedicated, enclosed, space to work in and conflict resolution techniques.

Develop Project Control Mechanisms: We try to minimize paperwork but our scope, schedule and lesson development documents identify project status at any time over 70 lessons, 25 steps for each lesson completion.

Maintain Documentation: All changes to our lessons are fully documented. Photo-reduced storyboards, printouts, lesson review comments are stored in lesson binders.

Update Authoring Tools: We have enhanced CTRAIN rather than compromise on simulation fidelity. For other authors this is not always, and often will not be, possible.

Feed the Environment: All lesson authors should have their theory of organizational development. Reactions to CAL are often bipolar in nature. It helps the cause to spend time informing colleagues, encouraging them to read storyboards, taking lessons, etc.

CONCLUSION:

Evidence is accumulating that CTRAIN lessons will be effective as one part of the operator training process for the Canadian Patrol Frigate Project.

My thanks for comments on the draft of this paper and for project participation are offered to: Paul Braden, Mark Garber and Bill Wylde of UNISYS, St. Paul, U.S.A.

Reference

1 Coleridge, S. T., The Ryme of the Ancient Mariner.

THE USE OF CAL FOR TECHNICAL TRAINING IN THE CANADIAN NAVY

P.H. Hope
Canadian Forces Fleet School
FMO Halifax
Halifax, NS

This paper will outline the historical background of the use of CAL in the Canadian Navy, present activities in the Westropp Laboratory for Computer-Based Training and the direction of CAL for the Canadian Patrol Frigate and Tribal Update and Modernization Programs.

The results of a trial aboard a naval destroyer and the findings of a research study will be presented, highlighting the successes of CAL in upgrading basic mathematics skills of naval technicians; thereby preparing them for higher level technical training required to maintain the new classes of ships.

UTILISATION DE L'EAO POUR LA FORMATION TECHNIQUE DANS LA MARINE CANADIENNE

Cette communication fait l'historique de l'utilisation de l'EAO dans la Marine canadienne et expose les opérations actuelles au laboratoire Westropp en matière de formation assistée par ordinateur, ainsi que l'orientation de l'EAO pour le Programme de la frégate canadienne de patrouille et le Projet de révision et de modernisation de la classe Tribal.

Les résultats d'un essai à bord d'un destroyer et ceux d'une étude seront présentés pour illustrer les succès obtenus avec l'EAO dans l'amélioration des compétences en mathématiques élémentaires des techniciens de la Marine, amélioration qui les prépare à recevoir la formation technique plus avancée indispensable avec les nouveaux types de navires.

THE USE OF CAL FOR TECHNICAL TRAINING IN THE CANADIAN NAVY

P. H. Hope

Canadian Forces Fleet School Halifax
Maritime Command
Halifax, Nova Scotia

In the middle of the 1970's, the Canadian Navy foresaw that the Navy of the 1990's would be technologically more advanced and that to use this technology effectively, new methods of training sailors would be required. Employment of the men of the Navy would also evolve as more advanced equipment was introduced into service. Computer-Aided Learning (CAL) was seen as a solution to training sailors for their jobs in the Fleet. From the early beginnings in the middle of the 1970's until the present day, CAL has undergone a gradual maturing and sophistication both in terms of fidelity and use. This paper will trace this evolution from the initial research projects through to state-of-the-art CAL systems for the new ships about to enter service.

Background

With the replacement of older ships by new patrol frigates and destroyers, come ships significantly upgraded in terms of both technology and sophistication. The changes in technology of the equipment in the Navy's ships necesitated a change in both the structure of employment of the sailors and the methods by which they were trained. To meet the needs of ship replacement programs, the Navy reorganized the men's trades from combined operator/maintainer trades into separate operator and maintainer/technician roles. To complete this reorganization a new training system evolved. As part of this reassignment, the technicians would be required to undergo a substantial academic upgrading program in order to prepare them for their entry level training and subsequent training.

When the Navy defined its training needs in the summer of 1978, its goals were to: make most courses modular; upgrade the level of technical competency for all technicians; conduct proficiency and threshold testing of personnel prior to long courses; minimize the movement of personnel between the Atlantic and Pacific coasts; and upgrade the academic level of all naval personnel. CAL was seen as one medium by which most of these goals could be met.

INITIAL Development Project

The Education Technology Project in the early 1970's demonstrated to the Canadian Forces that CAL was a viable alternative to more traditional methods of teaching. Project U-TRAIN was conceived as a research vehicle to explore both computer-based and paper-based individualized learning systems. The Navy was asked to participate in the design and implementation of CAL as a delivery medium for technical courses. Formalization of the specifications for the CAL portion of Project U-TRAIN soon followed and the Navy assigned responsibility for the development portion of this project to the Fleet School in Halifax, Nova Scotia. Further information on the results of this development project can be found in Kincaid and Saunders (1986).

CAN 8

While the development project was under way, it was decided to obtain a CAN 8 CAL system to provide remedial mathematics skills to the technicians about to enter the new naval training system. CAN 8 software from the Ontario Institute for Studies in Education (OISE) was seen as a solution. OISE CAN 8 CAL software was a low fidelity text-based system, with a high degree of modularity and feedback. Support for the CAN 8 system was forthcoming from the Commandant, Captain(N) Charles Westropp, who saw this CAL system meeting his requirements of preparing the best possible technician for service in the Fleet.

Initially, CAL was used as an adjunct to the regular classroom instruction for drill and practice of basic skills in mathematics. The use of CAL enabled the weaker students to improve their basic mathematics skills while not falling behind the rest of their coursemates. When the word was disseminated to the Fleet that the technician courses at Fleet School were difficult, the demand for terminal time by future students increased. Overall, the investment of approximately $120,000 (Cdn) per year for the CAN 8 system has proven itself by upgrading a large number of technicians who otherwise would have been forced to repeat their technical courses.

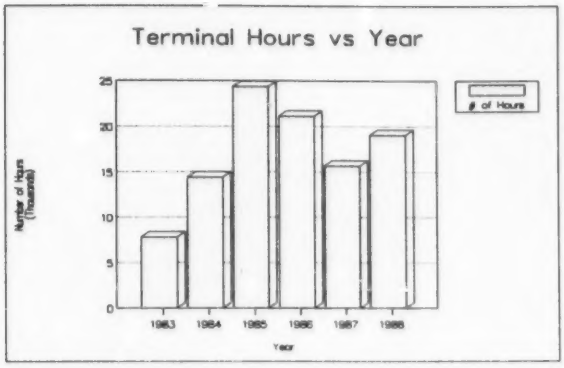


Figure 1

Expansions to the system soon followed, first with Modems to allow remote users to log on to the system during quiet hours, expansion from the initial 18 terminals to the present 54 terminals, leased lines to Canadian Forces Station Mill Cove and the Nova Scotia Teachers College, and the installation of a second system at the Fleet School in Esquimalt, British Columbia. Terminal hours have increased every year until now they are in the 18,000 hour range. Figure 1 shows the usage of the system for the past six years. Initially the system was heavily used. In the past year a physics course has been added, thereby broadening the range of courseware available to the student and usage is again up to approximately 20,000 hours per year. The 102,000 terminal hours represent the equivalent of 42 pre-academic courses 20 days long.

Remedial CAL Mathematics

Canadian Forces Personnel Applied Research Unit (CFPARU), in conjunction with the staff of Fleet School Halifax, conducted a study using CAL to replace some of the pre-academic courses offered by the School. The premise behind the study was that not all of the pre-academic students required the full pre-academic course. Students for the pre-academic courses were drawn from all facets of contemporary Canadian society, this resulted in a wide distribution of academic abilities. Traditional classroom instruction dealt with this wide scope by having the course start at some predefined level and then progress at a fixed rate. This often leaves students at both ends of the scale frustrated with the course and failures usually result for the weaker students. To this end a sample of students requiring the course were evaluated using the Canadian Achievement Test (CAT) for mathematical level. The results of the CAT measured the level of the student's ability in mathematics. These results were then displayed as an equivalent grade level. Based upon their CAT score, the students were then split into CAL and conventional instruction groups. The selection for each group was made on a random basis of students scoring the lowest on the CAT test. After the students did their respective training program, they were again tested using the CAT test. During the study the students were limited in the amount of time that they could spend on the CAL system for any given day. The results of the study showed that students taking the CAL program had a greater gain in CAT scores over the students taking the course via a conventional instruction. Table 1, from Halliwell 1986, summarizes the results of the study for the entire population.

Table 1
Mean Pre- and Post-Test CAT Scores for the CAL and Non-CAL Groups

Group	n	CAT Mean Scores Pre-Test	Post-Test	Gain
CAL	40	50.32	64.40	14.08
Non-CAL	39	47.82	51.82	4.00

The results of the study showed that the CAL group increased its CAT score by an average of 14.08 points, which is the equivalent of approximately three grade levels in mathematics. While the Non-CAL groups increase showed only a .75 grade equivalent improvement. The difference in the mean gain score is statistically significant at the .001 confidence level (Halliwell, 1986).

Another factor which was considered by the study was the threshold knowledge level of the students into the Qualification Level 5 (QL 5) training. For a technician this would be his first course involving intense academic study. The academic courses had been designed with an entry level equivalent to Ontario Grade 10 mathematics. The participants in this phase of the study all had an initial CAT score of less than 58, or the equivalent of Ontario Grade 10. The results in Table 2, from Halliwell 1986, show an even greater increase in the CAT mean scores, while the non-CAL group had approximately the same increase as the entire non-CAL sample population (Halliwell 1986).

Table 2
Mean Pre- and Post-Test CAT Scores for the CAL and Non-CAL Participants Commencing Training Below the QL 5 CAT Threshold

Group	n	CAT Mean Scores Pre-Test	Post-Test	Gain
CAL	31	47.52	63.16	15.65
Non-CAL	35	46.46	50.74	4.29

Shipboard Mathematical Refresher Training

The next stage in the evolution of CAL in the Navy was to evaluate the suitability of a microcomputer-based CAL system deployed onboard an operational ship. The Shipboard Mathematical

Refresher Training (SMART) program first sought to prove that microcomputer-based CAL was feasible and that the CAL platform be capable of operating in the demanding environment of a seagoing ship. This project saw the installation of a stand alone Plato system in a destroyer sized ship going on an extended deployment. To minimize the effects of the ship's motion on the system, it was placed in one of the stablest areas of the ship. Courseware for the system was selected to be equivalent to the suite of courseware for the CAN 8 installed in Fleet School. The only problems reported during this trial dealt with the labels on the discs and not with the system itself.

During the deployment of the ship a log was maintained of usage, weather and location. When the data from the log of student hours versus time was compared to the relative degree of sea state, as shown in Figure 2, it can be seen that the sea state did not have an effect upon the usage of the system, if anything it aided the usage of the CAL system. The 24 hour duty watch rotation and the self-paced nature of CAL meant that students could use the system at all hours. In a survey conducted by all of the users, it was generally agreed that CAL was beneficial for maintaining academic skills. The demand for time on the CAL system was not limited to technicians, but the number of interested students covered all of the occupation specialities employed aboard a warship. With the success of this initial trial, the Navy is now in the process of installing a CAL workstation in most operational ships.

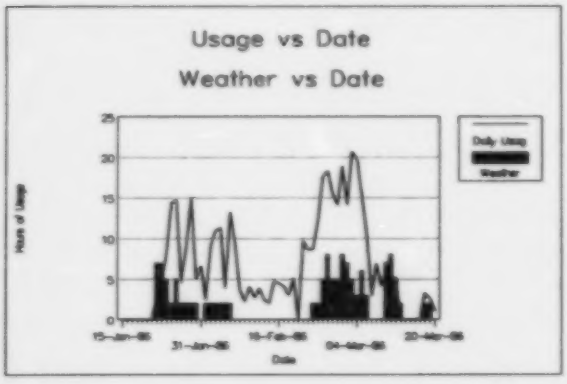


Figure 2

Future of CAL in the Navy

Within the next few years the Navy will be receiving two new classes of ships. CAL will play an important part in the training of the crews of Canada's future Navy. The Tribal class Modification and Update Program (TRUMP) for four of the Navy's destroyer size ships will see CAL being used extensively for both operator and technician training at all rank levels. The Canadian Patrol Frigate (CPF) program is in the process of defining the scope of CAL in the training of the crews of these new ships.

Concluding Remarks

CAL is an instructional technology that has already proven its worth to the Canadian Navy, and it will be used extensively in new equipment programs, in particular the TRUMP program. CAL systems have gone to sea in operational ships and have proven to the Naval training community that CAL is a reliable medium for upgrading the academic skills of the men prior to arriving on course. The installation of CAL systems on both coasts has ensured that the students are better prepared for their technical courses.

References

Halliwell, S.T. A Computer-Assisted Remedial Program on the Level of Mathematics Achievement Required for Naval Technical Training: Final Report. Canadian Forces Personnel Applied Research Unit, June 1986.

Kincaid, S.L. and Saunders, R.L. Requirements and Implementation Planning for CAL in the Canadian Forces. Proceedings of the Fifth Canadian Symposium on Instructional Technology, National Research Council Canada, May 1986.

LOGITEXTE: UN LOGICIEL DE CONCEPTION TEXTUELLE ASSISTÉE PAR ORDINATEUR

J.-Y. Frechette, R. Hamel
CÉGEP François-Xavier-Garneau
Québec, PQ

L'enseignement de l'écriture, tant pour les enseignants de la langue maternelle que pour ceux de la langue seconde, pose un défi de taille aux pédagogues. La problématique du logiciel éducatif ouvert ouvre cependant de nouvelles avenues dans ce qu'il conviendrait d'appeler: la pédagogie de la Conception Textuelle Assistée par Ordinateur. La conférence présentera les différents outils que LogiTexte met à la disposition des étudiants: bases de données lexicales, répertoires de structures syntaxiques et simulations rhétoriques. La présentation fera une large place aux exemples vécus lors d'une expérimentation avec des élèves de première secondaire dans la pratiques de production suivantes: a) le poème ludique; b) le texte informatif et c) le récit d'aventures.

LOGITEXTE: A COMPUTER-ASSISTED TEXT DESIGN SOFTWARE

The teaching of writing, both for teachers of the mother tongue and for second-language instructors, presents a major challenge to educators. However, the area of open-ended educational software provides new avenues in what may be called education by computer-assisted text design. This paper will present the various tools made available to students by LogiTexte: lexical data bases, lists of syntax structures and rhetoric simulations. Much of the presentation will be devoted to examples arising out of experiments with pupils in the first year of secondary school in producing (a) ludic poems, (b) expository writing, and (c) adventure stories.

LOGITEXTE: UN LOGICIEL DE CONCEPTION TEXTUELLE ASSISTÉE PAR ORDINATEUR

Jean-Yves Fréchette et Raymond Hamel

CEGEP François-Xavier Garneau

PROBLÉMATIQUE

La première génération de didacticiels conçus pour favoriser l'apprentissage de la langue maternelle furent des logiciels de type exerciseurs. Ces logiciels présentent successivement à l'écran une série questions suivies des réponses possibles parmi lesquelles l'élève doit indiquer quelle est celle qui lui apparaît la plus vraisemblable.

La pomme que j'ai_____

a) manger
b) mangé
c) mangée
d) manjé

Réponse: C

Le logiciel corrige les réponses de l'élève et peut même indiquer des pistes d'apprentissage en rappelant les règles de grammaires visées par la problématique des questions. Dans ce type de logiciel, l'élève intervient toujours dans le secteur limité prévu par l'exercice choisi et il ne peut s'en écarter ni dans les termes déterminés pour la réponse - qui se limite souvent à pointer ce qu'il croit être le choix juste - ni dans la thématique de l'exercice. C'est pourquoi on qualifie généralement ces logiciels de « logiciels fermés ».

UN LOGICIEL D'AIDE À L'ÉCRITURE

Or l'un des problèmes rencontrés par l'ensemble des professeurs de langue maternelle s'exprime par un constat navrant: les élèves écrivent peu et, quant ils le font, ils écrivent mal... Formulé autrement, on pourrait souligner que la grande lacune de l'enseignement contemporain est de ne pas permettre aux élèves d'exercer plus souvent leur compétence d'écrivain, de ne pas favoriser la multiplication des interventions écrites.

En allégeant considérablement toutes les opérations du traitement physique du matériau écrit, les logiciels de traitements de textes allaient quelque peu modifier ces perspectives. Ces nouveaux outils ont permis l'atteinte d'objectifs centrés sur la manipulation d'unités textuelles plus larges. Bien que stimulant la pratique de l'écriture, le traitement de texte n'avait pas permis l'apprentissage systématisé des notions fondamentales de la langue. C'est pourquoi la Direction des Ressources Didactiques du Ministère de l'Éducation du Québec nous a confié le mandat de mettre au point un logiciel de Conception Textuelle Assistée par Ordinateur: LogiTexte, outil dont la portée ne serait pas axée sur des fonctions de vérification des acquis linguistiques ou encore sur des fonction de correction du matériau verbal mais bien sur la manipulation du langage comme tel dans une authentique pratique de production textuelle.

PROTOCOLE D'EXPÉRIMENTATION

Pour répondre à ces attentes, nous avons alors privilégié le concept d'un logiciel où l'élève pourrait simultanément manipuler les deux composantes de tout acte de production de parole. Notre but était de condenser dans un protocole informatique simplifié les notions de sélection et de combinaison telles que définies par Roman Jakobson[1] et de les appliquer à des scénario de production textuelle.

Sur la base de ce modèle simplifié, nous avons d'abord proposé un premier prototype de logiciel à un enseignant de niveau secondaire[2]. Cet enseignant nous a vite fait part des attentes qu'un tel logiciel faisait naître chez lui: 1) ce logiciel devrait permettre à ses élèves de procéder à des acquisitions fondamentales dans le champs des notions de base en français; 2) ce logiciel devrait également pouvoir transformer cette connaissance en compétence par l'exercice et la pratique d'une écriture directement reliée à l'acquisition de ces savoirs de base: trois types d'écriture seraient ici privilégiés:

a) le poème ludique, b) le texte informatif et c) le récit d'aventures).

Il fut convenu entre nous que ce protocole de développement emprunterait une démarche qu'il conviendrait peut-être désormais de désigner comme un développement en contexte par prototypage. À chacune de la trentaine de rencontres que les auteurs de cet articles ont eues avec ce professeurs et ses élèves, le logiciel était testé, évalué et validé. Les élèves et le professeur faisaient alors les recommandations qui leur apparaissaient pertinentes et que les auteurs s'empressaient d'inclure dans le travail en cours comme autant de retouches successives qui viendraient s'ajouter au logiciel et aux exercices d'apprentissages qu'il rendait possible.

TROIS SAVOIRS FONDAMENTAUX

Cette approche nous a immanquablement conduit à la pratique d'une pédagogie de la langue maternelle dont l'apprentissage se fonde sur trois types de manipulations linguistiques. On pourrait alors parler de trois compétences distinctes:

1) acquisition de vocabulaire;
2) construction des schémas de phrases;
3) ajustement des mots aux impératifs de la structure syntaxique.

Il devient alors facile d'assimiler ces compétences aux trois types de savoir et de savoir-faire nécessaires à la maîtrise de la langue:

1) apprentissage des mots;
2) connaissance des règles de syntaxe;
3) usage simultané des deux axes dans le geste d'écrire: combiner les mots en fonction des priorités d'expression.

Cette dernière phase du processus d'écriture correspond à l'acte d'écriture comme telle. Il s'agit de combiner les mots dans des phrases signifiantes en y respectant tout autant les règles de la langue que ses propres exigences d'expression: ce que l'élève veut dire et comment il veut le dire. On voit bien ici que la nécessité interne du dire (les spécificités mêmes du code linguistique) et que les stratégies intimes de chacun pour signifier le réel et le nommer (les besoins d'expression des individus) se fondent dans ce qu'il convient d'appeler le style de l'expression. Ce style, jamais la machine ne pourra ni le commander ni le produire puisqu'il inclut dans sa définition même les variables contextuelles de l'expression aussi bien que les variables psychologiques de l'exprimant.

Pourtant, en se servant d'un logiciel de Conception Textuelle Assistée par Ordinateur comme support pédagogique, il est possible de penser que le pédagogue - le professeur de français en particulier - pourrait concevoir des modules d'apprentissage qui mènent à l'exploration a) d'un champ lexical organisé et b) d'une combinatoire syntaxique cohérente liés aux nécessités physiques et psychologique de l'expression écrite. L'élève, qui utiliserait un tel logiciel pourrait se soumettre aux exigences du code - et le respecter - et, en même, temps imposer sa stratégie d'expression. Un logiciel de CTAO permet aux variables expressives de trouver libre cours sans pour autant faire la concession d'un laxisme grammatical et structurale débridé: l'originalité ne s'acquiert pas au détriment de la qualité intrinsèque de la langue qui l'exprime. Expressivité et qualité linguistique se trouvent ici liées dans le texte généré par le médium informatisé. Il est donc possible avec un tel logiciel d'enseigner la créativité et de prôner l'originalité textuelle sans que la qualité proprement linguistique du discours n'en soit altérée. On ne sait que trop que les pédagogies qui mettent l'accent sur l'expressivité et la créativité le font souvent au détriment de la grammaire et des exigences formelles du code de la langue écrite.

Les enseignants qui pratiquent ces pédagogies misent davantage sur l'originalité des contenus plutôt que sur la qualité formelle de l'expression. L'utilisation d'un logiciel de CTAO comme LogiTexte permet de concilier ces deux objectifs. On a pu constater que les copies des élèves qui ont utilisé ce logiciel présentent des contenus d'expression originaux tout en affichant une tenue linguistique particulièrement exemplaire.

À cet égard, les trois thématiques stylistiques suggérées par le professeur (le poème ludique, le texte informatif et le récit d'aventures) nous ont permis de vérifier la souplesse du concept de CTAO tel que nous l'avons exploré avec LogiTexte. Qui eut pu dire qu'un même logiciel permettrait en effet d'explorer la richesse de l'univers poétique et immédiatement après servir d'outil privilégié dans la construction d'un texte qui ne souffre d'aucun écart par rapport au référent? Qui eut dit que le même logiciel puisse également servir de support dans la production d'un récit

d'aventures là même où l'imaginaire se déploie dans la fiction des aventures d'un personnage imaginaire?

REVOIR LES APPRENTISSAGES DE BASE

Cette démarche n'est possible que dans la mesure où elle assure la poursuite d'objectifs plus fondamentaux encore comme l'apprentissage des mots de vocabulaire et de leur organisation en réseaux sémantiques. Mais plus encore, il sera important que la pédagogie de la langue qui intègre la CTAO puisse permettre aux élèves d'accomplir simultanément des tâches d'analyse et de production. Autrement dit, il faut que l'écriture soit précédée ou accompagnée d'activités d'objectivation des éléments matériels de la langue. En ce sens, LogiTexte s'est avéré un outil exceptionnel pour que les élèves puissent objectiver dans le contexte de leur propre écriture les notions de lexique thématique, de réseau sémantique, de sens contextuel, de dynamique des accords grammaticaux et de grammaire de différents styles.

CONSTRUIRE SON UNIVERS LINGUISTIQUE

Avec LogiTexte, les mots - bien que suggérés par un processus de pige au hasard - ne font jamais que représenter l'univers linguistique de l'élève qui s'en est saisi puisque c'est lui qui, dans un premier temps, a garni chacun des fichiers lexicaux. De sorte que, s'identifiant instantanément aux mots suggérés par LogiTexte, l'élève peut alors se reconnaître - retrouver en quelque sorte la trace de son propre travail antérieur - à chacun des moments d'écriture. Dans ces conditions, le logiciel de CTAO agit comme un outil qui favorise le réinvestissement de ses propres matériaux linguistiques.

Utiliser un logiciel de CTAO comme LogiTexte dans les cours de français au primaire et au secondaire ne peut signifier autre chose que l'espoir de permettre à chacun des élèves d'atteindre rapidement une certaine maturité linguistique. LogiTexte, loin de mécaniser l'apprentissage, « responsabilise » au contraire l'élève en lui rappelant sans cesse que l'ordinateur est un outil d'aide qui ne peut aller au-delà de ses propres efforts et de ses propres investissements.

POLYVALENCE DES INTERVENTIONS D'ÉCRITURE

Structure vide que l'usager « meuble » au fur et à mesure de sa progression scolaire, LogiTexte agit comme outil d'apprentissage et d'exercices dans la manipulation du matériau linguistique écrit.

LogiTexte vise également à faire aimer l'écriture. Il va de soi que toute manipulation concrète du matériau verbal qui s'inscrit dans la poursuite des objectifs de production, ne peut faire autrement que développer un nombre important de pratiques connexes comme:

a) accroître son réseau lexical personnel
b) perfectionner ses modèles syntaxiques du code écrit de la langue
c) affiner ses stratégies personnelles de lecture et/ou de décodage
d) multiplier les activités « brouillonnage »
e) manipuler le jeu des interactions textuelles (perspectives stylistiques)

Il s'agit là d'un cadre général d'objectifs à atteindre pour quiconque désire acquérir une maîtrise adéquate d'une langue écrite depuis les tous premiers instants du primaire jusqu'aux derniers échelons du cours secondaire. Il conviendrait alors de voir comment ce logiciel-outil peut favoriser l'apprentissage des notions de base du vocabulaire des élèves tout en favorisant l'acquisition des habiletés fondamentales de la langue comme la maîtrise du schéma de la phrase simple par exemple en échelonnant ces apprentissages selon un système hiérarchique d'objectifs.

LOGITEXTE: UN LOGICIEL D'APPLICATION OUVERT

Comme tout logiciel éducatif ouvert, LogiTexte ne sera jamais riche que des efforts de chacun des élèves pour nourrir ses propres fichiers. A priori, il n'y aura pas dans LogiTexte d'exercices « tout fait d'avance », programmés par le maître ou le concepteur du logiciel. Il n'y aura pas non plus de banques de lexiques que l'élève pourra piller et utiliser comme bon lui semble. Tout doit être bâti par l'élève: les lexiques, les structures et les textes, bien sûr!

Tout dans LogiTexte est construit de telle sorte que l'élève puisse intervenir le plus souvent possible dans le processus d'écriture. LogiTexte ne permettra jamais autre chose que la reproduction exacte des compétences de l'élève. En fait, LogiTexte ne travaille qu'à partir des matières

premières qu'on lui apporte; voilà pourquoi il ne peut être autre chose que le reflet ponctuel de la compétence linguistique de chacun qu'il pourra, avec la complicité du maître, parfaire et bonifier.

C'est pourquoi, en ouvrant LogiTexte pour la première fois, l'élève sera devant une structure vide, une nouvelle structure ou un nouveau lexique. Pour parvenir à l'écriture, il devra faire ce que fait toute personne lorsqu'elle apprend sa langue: apprendre des mots, les répertorier dans des fichiers lexicaux personnels, formuler des schémas de phrases conformes à la syntaxe française et insérer au bon moment (dans les bonnes positions) les mots qui correspondent à la thématique du sujet traité.

CONCLUSION ET PERSPECTIVES

Il s'agit là d'une habileté fondamentale qui est partagé par chacun des usagers de la langue. C'est pourquoi nous croyons qu'un outil de CTAO comme LogiTexte pourrait être utilisé dans la production de textes dans le cadre de matières scolaires diverses: la géographie, l'écologie, les sciences naturelles en général ainsi que toute matière qui suppose l'utilisation d'un corpus spécialisé de vocabulaire pourrait tirer profit d'un tel logiciel en permettant aux élèves de manipuler dans le contexte de production de texte de recherche le vocabulaire propre à chacune de ces matières.

On pourrait même imaginer qu'un logiciel de ce type serve pour l'acquisition des notions de base dans l'apprentissage d'une langue seconde. Il serait alors utile de permettre aux élèves de pouvoir utiliser des mots nouveaux dans le cadre d'une production textuelle où serait privilégiée l'utilisation des modèles syntaxiques les plus courants.

Parce que LogiTexte permet d'isoler chacune des structures qui, d'une façon concomitante s'activent dans la langue (l'axe paradigmatique[3] [sélection du vocabulaire, perspectives morphologiques, constitution et élaboration des axes thématiques personnels, réseaux sémantiques] et l'axe syntagmatique [combinaison des unités lexicales entre elles, hiérarchisation des contenus, combinatoire expressive, dynamique grammaticale des accords]), ce didacticiel permet à l'élève d'intervenir à chacun des moments et à chacun des niveaux du processus d'expression et/ou de lecture.

Avec LogiTexte, l'élève peut ainsi multiplier les activités de manipulation du matériau linguistique aux niveaux morphologiques et syntaxiques; il peut, dans un texte donné comme structure de référence, isoler chacune des variables et les apprécier « en contexte » dans leur résonance sémantique littérale et/ou symbolique.

Conçu dans la perspective d'un outil de formation générale, LogiTexte répond aux besoins d'une pédagogie qui met l'accent sur l'apprentissage des habiletés fondamentales de langue maternelle. LogiTexte apparaît également comme un outil d'aide à la rédaction puisqu'il permet des pratiques pédagogiques dont l'enjeu est le développement de la créativité.

LogiTexte, enfin, pourra permettre aux enseignants pour qui la perspective littéraire s'impose comme un objectif d'apprentissage important, d'étendre leur propos pédagogique jusqu'aux visées stylistiques des variables contextuelles des différents genres littéraires.

---

[1] Roman JAKOBSON, Essais de linguistique générale, Paris, Éditions de Minuit, 1963. Jakobson y fait remarquer que l'acte de parole naît de deux actions simultanées: 1) choix/sélection des éléments lexicaux (mots) dans le réservoir personnel de chacun et 2) enchaînement/combinaison de ces différents mots dans la chaîne parlée de la phrase.

[2] Monsieur Bruno Laliberté est enseignant en Première secondaire à l'école Dollard-des-Ormeaux de la base de Valcartier près de Québec.

[3] Nous parlions plus haut de Jakobson qui, voulant définir les diverses mécaniques présentes dans le geste d'écrire, convenait de la présence de deux fonctions: la sélection et la combinaison. Jakobson représentait ces deux variables comme la rencontre d'axes perpendiculaires: l'axe paradigmatique qui désigne le réservoir de mots dans lequel puise le locuteur (le paradigme étant sa mémoire lexicale en quelque sorte) et l'axe syntagmatique surface où se combinent les différents éléments paradigmatiques. LogiTexte illustre schématiquement cette mécanique élémentaire: il pige des mots au hasard dans les différents paradigmes constitués par l'élève (ses lexiques) et les dépose à la surface de l'écran; ils s'articulent ensuite séquentiellement comme éléments d'une combinatoire dans la structure de la phrase.

CAL IN ARCHITECTURAL EDUCATION: MAKING THE DISTASTEFUL PALATABLE

J. Plume, K. Wyatt
School of Architecture
The University of New South Wales
Kensington, NSW, Australia

Most young people who wish to become architects do so because they have aptitudes in areas such as drawing, spatial perception, rhythm, pattern recognition and so on - the aptitudes that are reckoned to be centred in the right hemisphere of the brain. Unfortunately, the present-day practice of architecture, and especially the learning of architecture, requires great amounts of left-hemisphere activity: sequential, convergent thinking in the technologies of structural, mechanical and illumination engineering, environmental control, and acoustics. Many students of the artist-architect genre find these math-based studies highly unpalatable: they dislike the necessary rigour and distrust or devalue the technological approach.

The paper examines the use of microcomputers in architectural education, whereby we are meeting fire with fire: a fear of computation is being overcome by sitting the young artist before a medium resolution colour graphics display driven by a 16-bit personal computer. Three approaches are reviewed: individualized learning materials (traditional technology-based CAL in its various forms); architectural design aids (purpose-made interactive programs which perform the architectural/mathematical processes in a painless yet instructive way); and professional practice packages (programs designed for professionals in the workplace, used here in a learning environment). The importance of interactive graphics is stressed, and examples are drawn from several architecture schools in Australia.

L'EAO DANS L'ENSEIGNEMENT DE L'ARCHITECTURE: ADOUCISSEMENT D'UNE PILULE AMÈRE

La plupart des jeunes gens qui désirent devenir architectes choisissent cette profession parce qu'ils ont des aptitudes pour le dessin, la perception spatiale, le rythme, la reconnaissance des formes, etc., aptitudes qui ont leur siège dans l'hémisphère droit du cerveau. Malheureusement, la pratique de l'architecture de nos jours et plus particulièrement son apprentissage sollicitent grandement l'hémisphère gauche, lequel doit se charger de raisonnements portant sur les techniques de construction, le génie mécanique, l'éclairage, le contrôle de l'environnement et l'acoustique. Plusieurs étudiants du genre plutôt artiste trouvent ces études à base de mathématiques très rébarbatives: la rigueur nécessaire leur répugne et ils se méfient de l'approche technique ou la rabaissent.

Cette communication examine l'utilisation de micro-ordinateurs dans l'enseignement de l'architecture et nous attaquons le problème de front : nous aidons un jeune artiste à vaincre sa peur du calcul en le plaçant devant un écran graphique à définition moyenne dont l'affichage est commandé par un ordinateur personnel 16 bits. Examinons trois types de didacticiels : les didacticiels personnalisés (l'EAO traditionnel sous ses diverses formes), les didacticiels de conception architecturale (programmes interactifs spéciaux qui effectuent les opérations mathématiques de façon à la fois facile et instructive pour l'étudiant) et les progiciels professionnels (programmes conçus pour les professionnels dans leur milieu de travail et utilisés ici comme didacticiels). L'importance des programmes graphiques interactifs est soulignée et les exemples proviennent de plusieurs écoles d'architecture d'Australie.

CAL IN ARCHITECTURAL EDUCATION: MAKING THE DISTASTEFUL PALATABLE.

Jim Plume and Ken Wyatt

The University of New South Wales
Sydney, Australia.

EXPECTATION

Each year, more than eight hundred young people enter the fifteen recognised schools of architecture in Australia to fulfil their ambitions to become architects. Most of them come straight from twelve years of primary and secondary schooling. Although most have studied some maths during the final two years of school, a significant number have not studied physics during that period or perhaps have never studied it at all.

The entry requirements that the institutions place upon intending students vary widely. Eight schools have no specific pre-requisites other than matriculation. One school interviews all intending students who have performed satisfactorily in an aptitude test. Only three schools have maths as a pre-requisite and one has both maths and physics. Two schools warn intending students that prior knowledge of elementary physics and/or maths will be assumed.[1]

It can be seen from the educational background of the intending students and the entry requirements of the schools themselves that at least one perception is common to the two groups: you don't need much knowledge of maths or science to become an architect. This is probably a perception shared by the community at large: architects are seen as artists; manipulators of space, form and colour. A child who displays a talent for primary or even secondary school art is more likely to be advised by his parents to be an architect than a painter or sculptor. "Johnny is so artistic: we think he might like to be an architect when he grows up" is the prevalent, realistic attitude of fond parents who wish future financial security for artistic offspring.

REALITY

When Johnny enters architecture school, he finds that things are not like this at all. The courses of study in which students find themselves usually contain a strong technological segment, often amounting to a quarter or a third of the entire curriculum. There is usually a stream of structural engineering theory that will continue for three or sometimes four years. There will be a heavy commitment to environmental control: illumination engineering, heating and ventilating, solar control and acoustics. Construction processes and project management often require mathematical and statistical techniques. All of these technological studies involve mathematical modelling: many of them also involve the application of physical principles.

Although the mathematics required is certainly not of a very high order, all too often Johnny finds this whole approach very distasteful indeed. The 'Home Beautiful' magazines on his mother's coffee-table (which may have been his only background to architecture) said nothing about architectural technology. Probably the School's entry requirements either didn't mention mathematics at all or else suggested that prior knowledge was desirable but not essential. He came to the school because he was an artist; how could he possibly have anticipated that not only would he be expected to use applied mathematics, but, incredibly, he would be failed out of the school if he didn't apply it adequately.

Our observations in a number of countries show that this scenario is by no means unusual. Many students of architecture regard the study of the math-based subjects as (at best) an unpleasant and basically irrelevant chore or (at worst) an agonising, demoralising and thoroughly confusing experience. The faculty, having been students themselves in an earlier life, all too often share this perception. The prevalent attitude among the staff in many schools of architecture is (to place the kindest possible interpretation on it) ambivalent: they revere technology because they believe that by its use virtually any architectural fantasy can be realised, yet they believe that it is a lesser activity, best done by someone else, probably a rather grubby little man in a crumpled suit.

ARCHITECTURAL TECHNOLOGY AND COMPUTERS

What is the role of technology in the practice of architecture and therefore in architectural education? The essential aim of architecture, in the modern context, is to ensure that the end-product of the building procurement process is a building that fully serves its declared function in terms of usage, beauty, symbolism and so on. Architectural technology is concerned with the performance of the building fabric (brick, stone, glass, steel and plastic) which forms the raw material used by the architect to sculpt form and space so as to fulfil that aim. In order to properly utilise those materials and to create appropriate spatial form, an architect must be fully conversant with an ever-expanding field of

technological knowledge. Much of this knowledge is very specialised and firmly based in concepts of both maths and physics, so that the architect becomes dependent on specialist consultants to make proper use of those raw materials. Hence the popular assessment that architects are "jacks of all trades, but masters of none".

A major problem of architectural education is that in 5 or 6 short years (about 160 weeks or 3500 hours of face-to-face teaching), we must not only develop the student's creative skills in the sculpting of architecture, but we must also maturate their social and historical perceptions, provide them with the necessary graphic, written and oral communication skills, introduce them to business and contractual management processes and last, but not least, provide a sound basis for their on-going understanding of the technology of building throughout their professional life. Hence, the problems caused by the maths and physics components of the technology subjects for some students are exacerbated by the sheer volume of knowledge that is thrust in their direction.

The importance of the computer in architectural education lies in the way it enables students to gain an understanding of the technology of the building fabric *without the need to manually work through complex algebraic formulae*. We believe it is easier to appreciate the physics of a building by observational learning than by mathematical computation. For students who lack confidence in mathematical techniques, manual computation often merely obscures the underlying physical principle; the wood cannot be seen by a student who has to focus all his attention on the trees.. Increasingly, we must use the computational power (as well as the reasoning abilities, once that technology is established through AI research) of the computer to perform the calculation so that the student can observe the result and so comprehend the principle. In this way we fight fire with fire: we use a technology that is frequently feared by architects to demystify the concepts of architectural technology, both for the student and for the practising professional (who never really ceases to be a student).

An example of this approach may be found in the techniques involved in constructing a perspective drawing. Traditionally, these skills are perceived as a necessary part of architectural education because the designer needs to be able to illustrate his schemes for the client. In practice, this work is either handed out to a skilled specialist or handled by a computer. The educational merit of perspective construction is in developing the student's mental perceptions of form and space. This is best achieved using physical models, actually shaping the forms using tactile techniques. Unfortunately, that approach takes a great deal of time. Computers can supplement model-making by permitting a student to quickly experiment with alternative forms using computer graphic techniques. Spaces can be formed and modified and then explored by walk-throughs (impossible with physical models), colour experimentation (difficult with physical models), acoustic evaluation (also impossible with physical models) and lighting analysis (both visual and mathematical performance evaluation). The use of the computer in a traditionally artistic activity (drawing a perspective) can be extended into a series of perceptual and technological investigations, with a consequent saving in teaching time and without placing stress upon those students whose mathematical ability is not strong.

This example also demonstrates that, because the practice of architecture is an artistic activity, the graphical potential of the computer in itself becomes a way of reaching the student. We consider it important to use high resolution colour graphics so that the student is able to work in an attractive medium. Again, because architects rely so heavily upon visual means of communication, we consider it important that the software is well-designed graphically: "user-friendly" has a slightly different meaning to an architect than to a scientist.

The reality is that architects are not simply artists. They must have a *comfortable* understanding of building technology, yet the recognition of this has been sadly lacking in some of our schools in the recent past. The authors of this paper see the computer as a way, a *non-threatening* way, whereby the artistic student can be led to this sort of understanding. Where he can taste the technologies and hopefully find them to be palatable. As Michael Horowitz and his colleagues at Yale School of Architecture have put it: "Architecture is a profession that always seems to lag behind technology. We are constantly looking back in history for ideas, and then trying to bring them up to the present. Computers have to be brought into the schools of architecture so that we may be able to educate the new architects, the ones who will bring the new way out into the profession and convince others of the value computers presently have in the design portion of architecture."[2]

USE OF COMPUTERS IN TEACHING ARCHITECTURE

A recent survey (by the Committee of Heads of Australasian Schools of Architecture) has shown that computers are firmly established in the schools of architecture in Australia. All the schools have computers in the studio or in a laboratory and use them in the curriculum. In all cases there is time-shared access to a central mainframe, generally a DEC VAX minicomputer. Similarly, all schools have a number of PCs, generally IBM compatible, and commonly networked together as a laboratory. In most cases the number of PCs is somewhat more than half of the number of students admitted to first year.

The software in most schools in Australia tends to reflect the research interests of the staff. It will include the usual word-processing, database and spreadsheet packages found in commerce, together with one or two programming languages. It

will include a number of packages for the creation and manipulation of graphics ranging from simple "paint" programs to the 3D computer-aided design and drafting (CADD) systems. It will include a number of items relevant to that particular school's research thrust; some of these will be commercially-available packages, some will be ones developed in other architecture schools and some developed by the school itself.

There appear to be three different ways in which the computer is being used in architectural education in Australia: *individualised learning materials*, *architectural design aids*, and *professional practice packages*. The first of these is what Zinn described as "learning through the computer"; the other two would then be "learning with the computer"[3]. In the following sections, we will attempt an overview of each of these, together with a brief description of an example drawn from architectural education.

INDIVIDUALISED LEARNING MATERIALS

In this category would be included those traditional CAL packages that set out to teach the user some skill or concept at a user-defined pace, and which may include some progressive testing of the user's understanding as part of the process.

This traditional form of CAL is rarely used in architectural education, and we know of only one purpose-designed package, COMPOST. Some schools have made use of games which simulate parts of the process of building procurement, but even this technique is used most infrequently.

COMPOST (Computerised Principles Of Structures) was developed in the School of Architecture at the University of New South Wales in the early 1980s[4]. It consists of a suite of programs which covers the structural engineering theory usually taught in the first two semesters in an architecture course. It was designed for a Cyber 170/70 series machine, and was available to students through computing laboratories across the campus.

COMPOST consists of six lessons, covering six topics, commencing with equilibrium of forces at a point, and concluding with structural design of a timber beam. Each lesson commences with a brief review of the subject matter and then seeks step-by-step input from the user to solve two or three exercises, before setting several homework problems, different for each user. At the next lesson, the homework is checked, the results are recorded and explanations are offered for problems not correctly solved. COMPOST's most beneficial attribute is this ability to "diagnose" the input so as to identify the error made by the student; the suite's designer was both a fine engineer and a dedicated teacher, and his ability to predict the variety of input was uncanny. COMPOST was rather less successful as a learning manager; it suffered a number of problems related to the use of the University's mainframe and the multiplicity of laboratories used. Dealing with the CPU eventually became so irksome the we ceased using COMPOST; it is now more than time that we revised it for use in the School's PC laboratory.

ARCHITECTURAL DESIGN AIDS

In this category would be included interactive packages that enable a user to examine, test or evaluate some aspect of the building's design for the purpose of improvement. Usually, these programs have been developed by teachers in response to needs perceived in the studio; the programs are intended specifically for the student or inexperienced designer, and provide help usually during the design development phase of the design process.

The most common application area for the development of design aids is environmental control. One school (Newcastle), for example, has produced packages for solar heat load, climate control, sunlight penetration, domestic lighting calculation and reverberation time. Four or five other schools have produced similar suites of programs. Planning and layout problems have also received a good deal of attention, as has perspective rendering. Although these packages are exchanged between schools, few of them are commercially available as yet.

SIZE is a design aid currently under development by one of the authors. It is an interactive, menu-driven suite of programs which provides an estimate of the cross-sectional size of structural components such as beams and columns. The objective in designing SIZE was to produce an architect-friendly program able to provide "guestimates" which would be at least as accurate as those provided by consulting structural engineers to architects in practice. The user part of the package is totally devoid of mathematics; in fact, in most applications only two numeric inputs are required. The package is intended for studio use on a PC, preferably fitted with an EGA card.

Another example of a design aid program developed in our School is called MODEL[5]. Its purpose is to enable students to better appreciate the spatial form of their buildings during design studio exercises. It runs on an IBM PC AT compatible using EGA graphics. The student begins by constructing the plan shape of his proposed design by stringing wall elements together, inserting windows and doors, and placing a roof form over the structure. Because each element has user-defined height data associated with it, the program is able to construct a 3D model from that plan data and display a perspective image of the result. The program allows students to make simple changes to their model and then produce relatively fast hidden surface views of the building.

This program is more than just a perspective program. If used properly, it becomes a powerful learning aid to permit students to appreciate the spatial form implications of their planning decisions when laying out the spaces in their

buildings. This is a crucial skill for architects which is not always easy to develop. MODEL provides the kind of rapid 3D feedback that encourages the student to explore different possibilities.

It must be apparent that these design aids are not CAL packages in the conventional sense in that they do not set out to teach a skill or principle. However, they do facilitate the process of architectural design, and that is a central concern in the schools.

PROFESSIONAL PRACTICE PACKAGES.

Professional practice packages represent the third category. These are packages designed for use in the commercial practice of architecture and the related professions, and were never intended for pedagogical use. Yet we have found that, when used selectively and with clear intent, they have a valuable place in the teacher's armory, particularly when used by senior students in elective classes.

The most widely used packages are those for computer-aided design and drafting, usually 2D but occasionally 3D. The usual basic university CADD course is vocation-orientated, and has little to do with CAL. However, it is possible to use CADD as a medium for exploring the spatial assembly of buildings that are recognised examples of good architecture. In that way, the student's appreciation and understanding of how to manipulate space is vastly enhanced. Rather than merely teaching the student how to use a CADD system (analogous to teaching manual drafting skills), students are able to develop the skill while using the tool to enhance their understanding of building form and geometry.

Similarly, other packages which deal with air-conditioning, illumination design, project management, network analysis, thermal transfer (not to mention the commercial workhorses like spreadsheets, databases etc.) can all be used to remove the drudgery of manual computation from a student's investigation, and leave the principle exposed.

MICROSTRAN (distributed by Engineering Systems Pty Ltd, Sydney) is a package designed for and used by consulting structural engineers. It is a system of programs which enables the analysis of three-dimensional structures of a very wide variety of types, and which can then proceed through to final design of the components, ready for manual or computer-aided drafting. It is a very powerful engineering package, and certainly not one that was ever designed for use by students of architecture. Yet we chose it for use by senior architecture students in the engineering elective subjects, because it then becomes possible to model the behaviour of a range of so-called "statically indeterminate" building structures which are far too difficult for study by hand computation.

We use MICROSTRAN in a series of "investigations", in which the student makes rapid comparisons between different structural systems, commencing with those with which he is thoroughly familiar, and moving quickly into completely foreign territory. The investigations pose a sequence of questions which require the student to consider *the physical behaviour of the real structure* represented by the mathematical model, and then to modify that structure in the light of discoveries made in response to earlier questions. Even though this approach is used only in the elective subjects, we have found that even those students who lack confidence in their mathematical background have been able to come to a new understanding of structural behaviour, once the mathematics has been removed by use of the computer.

CONCLUSION

Architecture schools are using computers to make unpalatable mathematics-based topics both acceptable and understandable to students. Although little use is made of the individualised learning packages that are the mainstay of CAL in other disciplines, appropriate computer-based educational techniques are being devised and adopted. These include both purpose-written software that is used to aid students in their undersatnding of principles and techniques in a wide range of disciplines, as well as commercial, profession oriented packages which can be used to teach architectural principles while giving the student an appreciation of marketplace techniques.

REFERENCES

1. Holden, G.,ed.: Architecture Schools of Australia 1988, Royal Australian Institute of Architects, Canberra, 1988

2. Horowitz M., Gehner M., & Loomis C.: Computers in the Yale School of Architecture, Proceedings of the 1985 IBM Academic Systems University AEP Conference, IBM, Connecticut,1985, pp133-136.

3. Zinn, K.: An overview of current developments in computer-assisted learning in the United States. *Programmed Learning and Educational Technology*, 15,2,126-135.

4. Hough R.: Teaching statics of structures by interactive computer program, *Architectural Science Review*, June 1984.

5. Peter, S.: Modelling and Perspective Programs in Architectural Design and Education, Ausgraph88 Proceedings, July 1988, pp. 313-318.

THE LEARNING CENTRE - A NEW SOLUTION TO STAFF DEVELOPMENT CHALLENGES

G.V. Burston, C.G. Crapper
Corporate Training & Development Department
Ontario Hydro
Toronto, ON

Employee training and development efforts have traditionally faced several challenges: the high cost of workshops, seminars and conferences; squeezing training programs into hectic staff schedules; excessive delays between enrollment in a program and attendance; expensive transportation and accommodation costs; providing "in house" or external training to a large diversified staff group.

The Learning Centre is a self-paced learning facility that provides a large variety of highly cost-effective training programs while virtually eliminating travel and accommodation expenses. Wait times are reduced from weeks or months to days or even hours. In addition, the Learning Centre allows staff to schedule training around their timetable and reduces the time invested in training.

As a proven technology, Ontario Hydro's Corporate Training & Development Department has been successfully operating their Head Office Learning Centre for four years, with subscriptions growing annually. (In 1987, 2,064 participants used the Centre.) The combination of flexible methods (computer-assisted, video-based, etc.) and high quality software has led to this being a widely-accepted mechanism for delivery of non-technical training which is difficult to schedule using more traditional strategies.

LE CENTRE D'APPRENTISSAGE - UNE NOUVELLE SOLUTION AUX VIEUX PROBLÈMES PERFECTIONNEMENT DU PERSONNEL

Les responsables de la formation et du perfectionnement du personnel font face, depuis longtemps, à plusieurs problèmes: le coût élevé des ateliers, des séminaires et des conférences; l'intégration des programmes de formation à l'emploi du temps déjà chargé des employés ainsi que les délais excessifs entre l'inscription et la présence à un programme; des frais de transport et de séjour élevés; la formation dans l'entreprise ou à l'extérieur donnée à un groupe nombreux et diversifié.

À rythme libre, le Centre d'apprentissage offre un large éventail de programmes de formation très rentables qui éliminent, en fait, les frais de déplacement et d'hébergement et diminuent les temps d'attente, qui passent de semaines ou de mois à des jours ou mêmes des heures. De plus, ce Centre permet au personnel de planifier sa formation en fonction de son emploi du temps et réduit le temps consacré à la formation.

La formule ayant fait ses preuves, le service Formation et Perfectionnement internes d'Hydro Ontario dirige, avec succès depuis quatre ans, le Centre d'apprentissage du siège social, à l'aide de cotisations qui augmentent chaque année. (En 1987, 2064 participants ont eu recours aux services du Centre.) Grâce à ses méthodes souples (on y utilise l'ordinateur et la vidéo) combinées à des logiciels de grande qualité, le Centre est devenu un instrument largement reconnu pour transmettre une formation non technique, difficile à planifier avec des méthodes plus traditionnelles.

THE LEARNING CENTRE: A NEW SOLUTION TO STAFF DEVELOPMENT CHALLENGES

G.V. Burston, C.G. Crapper

Corporate Training and Development Department
Ontario Hydro
Toronto, Ontario

INTRODUCTION

Ontario Hydro is one of the largest and most efficient electrical utilities in the world, supplying electricity to eight million people in the Province of Ontario and covering a geographical area of 412,500 square miles. (In comparison, the state of Texas, USA, covers 267,5000 square miles.) In addition, Ontario Hydro exports electricity and is involved in many international mega-projects in Africa, Asia and South America.

Emergent social, geographical, technological, economic and political factors have a major impact on the administration and operation of this large, multi-functional corporation of 24,000 employees. Challenges range from how to manage a diverse workforce (e.g. the implementation of employment equity and affirmative action programs) to the management of sophisticated information technology, to the politics of nuclear power, acid rain, etc.

Training plays a critical role in the maintenance of Ontario Hydro's "leading edge" and a fast response to changing needs is vital. This paper presents empirical data to support the role of instructional technology in meeting at least some of the demand for non-technical training, via a "Learning Centre" approach.

THE CHALLENGE

As a result of our rapidly changing society, the need for an ongoing and flexible approach to management development, supervisory training and professional skill development has become increasingly important. Like many large corporations, a major challenge faced by Ontario Hydro is how to provide managers and supervisors with the skills and knowledge required to manage their staff in a changing society. In addition, how does an organization continue to upgrade the work management skills of its professional staff in areas such as time management, problem/decision analysis, stress management, etc?

Some of the major problems faced by Ontario Hydro's Corporate Training and Development Department in the past have included:

1. The demand for training exceeded the supply capability.
2. Fixed workshop dates did not accommodate busy staff schedules.
3. The cost of training (travel, accommodation, lost opportunity, etc.) put a strain on development budgets.
4. The need for follow-up or refresher training was not being met.
5. Staff and their supervisors wanted to minimize time away from the job.
6. Clerical and administrative staff did not get sufficient training.

So, responsiveness to client needs, timeliness of training, program quality, and training costs all represented challenges to Ontario Hydro's delivered system for management and professional development.

One-off management training programs and refresher sessions every few years became insufficient to meet these challenges. Ontario Hydro needed a flexible delivery strategy which would enable managers, supervisors and professional staff to satisfy their ongoing needs without resorting to costly external programs or waiting until the Corporate Training Department put on a course which might, or might not, address the right issues and might, or might not, be available at the right time.

A number of solutions to these problems were considered. These included: increasing the frequency of formal classroom training; the use of externally delivered programs; the use of new technology to deliver training (computer, videodisk, etc); access to formal educational programs offered by the universities, and so on.

A SOLUTION - THE LEARNING CENTRE?

In 1985, in response to these challenges, Ontario Hydro decided to continue to use a strategy that centred on formal classroom-delivered management and professional courses, but supplemented by a carefully selected range of packaged, self-paced programs that form part of a training software library delivered via a "Learning Centre". It was decided that a combination of self-paced text, audio-cassette, video- and computer-based materials would be included.

A Performance-Based Approach: In keeping with Ontario Hydro's philosophy of performance-based training, a front-end analysis was carried out which examined the magnitude of the problem and identified major areas of skill/knowledge deficiencies. Courseware would be selected based on these needs in order to allow the Corporate Training and Development Department to control the overall quality of the management and professional training in the Corporation. Logistical issues which needed to be addressed included:

1. Space limitations in an already overcrowded Head Office building.

2. Selection and cost justification of the appropriate hardware for delivery (computers, videotape/disk equipment, etc.).

3. The availability of high quality courseware (which was very limited at the time of the pilot).

4. The capability of staff to select quality programs and link these with the high-tech delivery media.

5. Advertising and promoting the programs in the head office environment.

6. The availability of administrative staff to run the Centre.

7. Close monitoring of the use and impact of the Learning Centre.

8. Maintenance of a library of courseware combining purchased materials (to satisfy ongoing training needs) with materials rented on a rolling basis (for special, short-term issue related situations).

The Pilot: The criteria for the success of the Learning Centre included: flexibility in meeting individual needs; accessibility to the Centre and courseware availability; administrative efficiency and effectiveness; and, of course, program quality and cost effectiveness. A pilot took place from September 1985 to February 1986. Ontario Hydro's Head Office in Toronto was selected as the pilot site because the majority of the Corporation's management and professional staff work at this location (approximately 7,000 people including clerical staff).

Participant Feedback: Results of the pilot indicated that the Learning Centre concept was, indeed, viable as a strategy for supplementing the more formal classroom delivery for management and professional development. Feedback from participants on the pilot program indicated that their needs for "training on demand" were being met to a large extent by the Learning Centre.

Stakeholder Feedback: From the Training Department's point of view, the Learning Centre concept seemed to have potential because the Department could then control access to the materials available in the Centre, thereby controlling quality and addressing skill and knowledge deficiencies directly. Line Management was positive, because it provided their staff with the right training at the right time, without staff being away from their work location for extended periods of time.

OPERATING EXPERIENCE

Since the pilot four years ago, use of the Learning Centre has grown rapidly. The Centre provided 729 person days of training in 1987 rising to 843 person days in 1988. This translates to over 2,000 programs in each year for approximately 1,600 staff per year. This demand is about 25% below present capacity which can be increased quickly with the addition of more work stations. Broader-based Corporate demand from field sites is being met by local Learning Centres.

Over the past four years, our experience indicates that, for management and professional development:

1. The Learning Centre is an important strategy for the delivery of training when used in combination with more structured approaches; self-paced strategies do not lend themselves so well to situations where interpersonal or group interaction skill development is required.

2. The selection of hardware is crucial, particularly in the case of computer-based delivery, where the hardware and operating system must be compatible with the courseware available.

3. The amount of high quality courseware available for self-paced management and professional development has increased during the past four years, particularly in the areas of CBT and video-based programs. Software and system compatibility has also increased, so that a wider variety of courseware is available for our specific (largely IBM) equipment.

4. The selection of high-quality courseware tied to real needs is essential to achieve impact. This means that continuous monitoring of client needs and available materials must occur simultaneously, i.e. programming the Learning Centre as a dynamic, not a one-time activity.

5. An efficient system of nomination and administration is vital to the success. The day-to-day operation of the Centre is carried out by an Administrator, well trained in the nomination procedure, the operation of the equipment, and who has excellent customer service skills.

6. Promotion of the concept and advertising is important. People will be more inclined to use the Centre if they are comfortable with the technology. Lack of familiarity with computer technology did not present a significant problem, mainly due to the comparative sophistication of the users and the way in which the administrative staff are trained to help troubleshoot operating problems. Informing potential users of program availability is an ongoing problem, since the Learning Centre changes its programs every three months. Advertising strategies include posting the upcoming calendar on bulletin boards and distribution of the calendar to Human Resource Departments and Personnel Offices throughout the Head Office location.

7. The cost of delivering management and professional training has been reduced significantly using the Learning Centre strategy and flexibility has increased. Some training that would have taken place in the classroom is now available in the Learning Centre. Where content and quality of the two strategies is similar, cost data collected during the past four years demonstrates that the Learning Centre is the more cost-effective alternative. Our data shows that the cost per participant day for the Learning Centre is approximately 45% of the cost per participant day for classroom instruction.

SUMMARY

Ontario Hydro's experience with its Learning Centre has demonstrated that computer-assisted and video-assisted techniques for the delivery of non-technical training is a viable option in meeting the challenges faced by today's organizations. Depending on the application the instructional technologies can, in some cases, meet the primary delivery requirements for training (i.e. they can stand alone to fulfill training needs) or they can supplement other, more traditional, techniques for training.

Experience also shows that the flexibility in meeting client demands has increased while, at the same time, there has been a significant reduction in the participant day cost for the delivery of non-technical training.

COURS DE FRANÇAIS ASSISTÉ PAR ORDINATEUR (EAO) À L'INTENTION D'UN GROUPE DE FONCTIONNAIRES

S. Beauchemin
INFOJED
Montreal, PQ

Cette communication veut décrire la démarche de conception et de développement d'un cours de perfectionnment en français écrit assisté par ordinateur, à l'intention des fonctionnaires de Revenu-Canada. Ce cours a comme objectif l'amélioration de la performance en français écrit du personnel du Ministère dans le domaine de la correspondance interne et externe.

La ressource informatique nous a permis d'organiser ce cours dans une perspective d'individualisation, en fournissant aux usagers des rétroactions appropriées à leurs réponses et en leur permettant de choisir, à travers des "menus", le procédé pédagogique qui leur convient (lectures de règles, d'exemples, questions de support, questions de révision). Des tests sont aussi prévus dans le but de fournir à l'usager un diagnostic de sa performance.

Cette réalisation se situe dans le cadre général des applications pédagogiques de l'ordinateur et, en ce sens, présente un intérêt autant pour les intervenants de ce domaine que pour les observateurs qui souhaitent en suivre l'évolution.

COMPUTER-ASSISTED FRENCH COURSE (CAL) FOR A GROUP OF PUBLIC SERVANTS

The purpose of this paper is to describe the procedure for designing and developing a computer-assisted development course in written French for employees of Revenue Canada. The goal of this course is to improve the performance of Department employees in writing French for internal and external correspondence.

The computer resources available enabled us to organize this course with a view to individualization, providing users with appropriate feedback to their responses and allowing them, through menus, to select the learning process that suits them (display of rules, examples, supporting questions and review questions). Tests are also planned to provide users with a diagnosis of their performance.

This project comes under the general heading of computer applications to education, and thus will be of interest both to those involved in this field and to observers who wish to follow its development.

COURS DE FRANÇAIS ASSISTÉ PAR ORDINATEUR À L'INTENTION D'UN GROUPE DE FONCTIONNAIRES

Suzanne Beauchemin

Infojed

Ce cours, dont l'élaboration s'est amorcée en août 1988, est destiné à un groupe de fonctionnaires fédéraux. Le contenu du cours devait être adapté à cette clientèle à plus d'un niveau : nous avons donc procédé à une analyse de besoins qui a orienté la forme et le contenu du didacticiel. Ainsi, ce groupe de fonctionnaires a pour caractéristique de travailler en milieu bilingue ce qui a, bien sûr, des incidences sur les choix de thèmes grammaticaux à aborder dans un cours de français. De plus, le contenu devait rester intimement lié au contexte particulier de ce milieu de travail, notamment au niveau du choix des exemples illustrant les règles autant que du choix des "problèmes" linguistiques traités. Les trois objectifs qui suivent ont déterminé la démarche de conception et de développement de ce projet; ils se sont précisés à la lumière de l'examen de documents provenant du milieu de travail lui-même.

Objectif linguistique

Notre didacticiel a comme premier objectif l'amélioration de la performance en français écrit dans le domaine de la correspondance interne et externe de ces fonctionnaires. Il s'agit donc d'un cours de perfectionnement en français écrit et couvre deux grands aspects de l'écrit, à savoir : la grammaire proprement dite et la rédaction. Le volet touchant la grammaire tente de couvrir les principales règles d'accord et d'utilisation du lexique; celui sur la rédaction propose une révision des principes de l'écriture efficace. Le lecteur trouvera dans les pages qui suivent une description plus détaillée du contenu du cours.

Objectif pédagogique

Nous avons tenu à organiser ce cours dans une perspective d'individualisation, comme la ressource informatique le permet avec, cependant, les limites de celle-ci. La présence des menus assure le libre déplacement de l'étudiant à travers les activités du cours et lui permet le choix de tâches adaptées à ses besoins. Ces menus offrent aussi à l'étudiant la possibilité de choisir le procédé pédagogique qui lui convient (lecture de règles, d'exemples, questions de support, questions de révision).

Nous pensons, d'une façon générale, que la présence de rétroactions aux réponses de l'étudiant constitue une condition essentielle à l'individualisation. Nous avons donc tenu à ce que cet aspect soit présent dans ce didacticiel : ainsi, l'étudiant reçoit une rétroaction appropriée à chacune de ses réponses et lui expliquant la source de son erreur, s'il y a lieu.
Le cours en tant que tel est précédé d'un pré-test et suivi d'un post-test pour lesquels l'étudiant obtient une note. Ainsi, l'étudiant pourra identifier ses besoins à l'aide du diagnostic de départ et mesurer ses acquis par le diagnostic d'arrivée. Toujours dans une perspective d'individualisation, le cours lui-même n'est pas séquentiel, c'est-à-dire que les activités ne sont pas proposées dans un ordre absolu : l'étudiant peut choisir à la fois les activités qui l'intéressent et l'ordre dans lequel elles lui apparaîtront.

Objectif médiatique

Toute la médiatisation s'est organisée moins sur la base de nos propres choix qu'à partir des possibilités que nous

offraient le système central du Ministère à qui le didacticiel est destiné. Ce cours de français est donc médiatisé en Phoenix, un système-auteur qui constitue l'encadrement informatique à partir duquel nous avons travaillé et que nous avons tenté d'exploiter au maximum. Toute la conception du cours est donc fortement déterminée par les possibilités de Phoenix : ainsi, l'analyseur de réponses traite les choix multiples, les réponses en oui/non, les réponses déjà prévues en conception, mais ne permet pas de réponses dites "ouvertes", c'est-à-dire non-prévues.

Il va sans dire que ces trois niveaux d'objectifs sont étroitement interreliés dans une telle démarche à tel point qu'il est parfois difficile de distinguer, dans nos choix, ce qui est pédagogique, ce qui est linguistique ou médiatique.

Description du didacticiel

Le cours de français est constitué de deux volets : Perfectionnement en grammaire et Perfectionnement en rédaction. Chacun de ces volets se divise en modules, destinés à une thématique grammaticale. Ainsi, le volet "grammaire" contient sept modules portant sur :

a) Une révision des bases de l'analyse grammaticale et logique.
b) L'accord du verbe avec son sujet
c) L'accord de l'adjectif qualificatif
d) L'accord du participe passé
e) Majuscules, ponctuation, sigles, acronymes
f) Les termes d'articulation du discours : les mots-liens.

Le volet "rédaction" contient quatre modules :

a) Les principes de l'écriture efficace
b) Les niveaux de langue
c) La lettre
d) Problèmes de structures de phrase

Répétons que le cours est précédé d'un pré-test à chacun des modules, où l'étudiant doit répondre à une dizaine de questions tirées de façon aléatoire d'une banque qui en contient quarante. Un diagnostic commenté est fourni à l'étudiant qui décide alors s'il veut aborder le module correspondant au test ou pas. Le post-test est construit de la même façon et son diagnostic suggère à l'étudiant de reprendre le module si le seuil de réussite n'est pas atteint.

Chaque module est précédé de la présentation de ses objectifs, de la description de son titre et d'un menu général qui offre à l'étudiant un choix d'activités : le contenu lui-même du module ou une série de questions portant sur la matière de ce module et appelées "questions de révision". Le module présente les règles liées à son contenu grammatical; ces règles sont illustrées d'exemples et accompagnées d'au moins deux questions auxquelles l'étudiant répond et pour lesquelles il obtient une rétroaction lui expliquant son erreur, s'il y a lieu. Le contenu du module offre aussi des menus quand la matière peut se segmenter : par exemple, c'est le cas du module portant sur "L'accord du participe passé" où l'étudiant peut choisir entre diverses règles d'accord du participe passé avec l'auxiliaire "être", l'auxiliaire "avoir", les pronominaux, etc.

Les "questions de révision", une vingtaine en général, fournissent aussi des rétroactions indiquant à l'étudiant s'il a bien compris la matière du module qu'il vient de compléter ou alors, lui indiquant qu'il devrait choisir l'activité "contenu du module", s'il ne l'a pas fait. Le temps requis pour chacun des modules varie de 30 à 60 minutes, selon la complexité du contenu.

Nous avons porté une attention particulière à l'organisation des écrans; nous avons souhaité éviter la surcharge au niveau du texte, comme au niveau de la mise en évidence (souligné, surbrillance, graphisme, etc.). Les écrans sont donc aérés, présentent brièvement la matière et les questions posées systématiquement donnent un rôle plus actif à l'étudiant que la simple lecture de règles, tout en lui permettant de tester sa compréhension.

Matériel d'accompagnement

Au moment de son inscription au cours, l'étudiant reçoit un "Manuel du participant", décrivant en détails le contenu et la forme du cours. Toutes les règles ainsi que les exemples qui les illustrent y apparaissent : le Manuel sert donc à la fois d'introduction au cours et d'aide-mémoire

conservant la matière et facile à consulter. Nous y avons inclus un lexique, reprenant et définissant les termes grammaticaux utilisés dans le cours; cet outil peut être bien pratique pour des utilisateurs qui n'ont pas vu depuis longtemps le mot "conjonction", par exemple! Nous y avons aussi inclus une série d'exercices complémentaires, avec leur corrigé, portant sur des problèmes plus fréquemment rencontrés chez nos étudiants et plus difficilement traduisibles pour Phoenix, compte tenu de leur complexité syntaxique.

L'équipe de travail qui a assuré la conception, le développement et la médiatisation de ce cours était constituée de quatre personnes, soit, un fonctionnaire qui nous a fourni les documents nécessaires au choix du contenu, une linguiste, chargée de la recherche au niveau des règles et des exemples, une linguiste, spécialisée en A.P.O., qui a développé le cours et une informaticienne qui a médiatisé en Phoenix. Un tel travail exige, il va sans dire, l'énergie et la compétence de plusieurs personnes et est impossible sans une collaboration de tous les instants entre les membres de l'équipe.

THE ROLE OF LOGIC IN COMPUTER-ASSISTED INSTRUCTION

J. Skosnik
Industry Services Program
British Columbia Institute of Technology
Burnaby, BC

Since May, 1986, our research group has been developing industrial applications for artificial intelligence, particularly as these applications relate to advanced industrial education. Over-all, our work is an attempt to deal with three pressing problems in advanced industrial training, namely: 1) the need for on-site hardware specific training; 2) the high cost of modern industrial hardware, which severely limits what can be done in conventional training laboratories, and 3) the need to include more intelligence in computer-based instruction, both for user-system interactions and for truth maintenance in the changing knowledge base.

Accomplishments to date include: a) the development of a sound and complete inference engine which will be used to manage the flow of information in a hypertext tutorial program, and b) the development of simulation software for a large class of important industrial hardware (Programmable Logic Controllers). Considerable technical difficulties were overcome in the course of developing the sound and complete inference engine, which is required to insure that the AI component in the learning system fulfills the requirements of mathematical logic. The resulting system, while highly interactive, is logically incorruptible, since it is impossible to extend the knowledge base in contradictory ways. Other logical checks on the system also prevent users from adding logically redundant information to the knowledge base; this greatly improves program execution. There are plans for an on-site implementation of the computer-based learning system which will arise out of our work. This will occur in British Columbia at Prince George Pulp and Paper Mills as part of a 22 million dollar plant modernization effort.

LE RÔLE DE LA LOGIQUE DANS L'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR

Notre groupe travaille depuis mai 1986 à la mise au point d'applications industrielles de l'intelligence artificielle, particulièrement des applications touchant l'enseignement supérieur des techniques industrielles. Par dessus tout, nos travaux visent à répondre à trois besoins pressant en formation industrielle de niveau supérieur, soit : 1) formation sur place sur l'utilisation d'un certain matériel, 2) formation adéquate en laboratoire en dépit des contraintes imposées par le coût élevé du matériel industriel moderne, et 3) addition d'une plus grande part d'intelligence dans l'enseignement assisté par ordinateur, autant pour les besoins des interactions utilisateur-système que pour la véritable tenue à jour d'une base de connaissances évolutive.

Parmi les réalisations à notre crédit, mentionnons A) la mise au point d'une machine à inférence complète et sonore qui sert à gérer la circulation des informations dans un tutoriel hypertexte, et B) la mise au point d'un logiciel de simulation destiné à une grande catégorie de matériels industriels importants (Contrôleurs logiques programmables). Nous avons vaincu d'énormes difficultés dans le développement de la machine à inférence, surtout pour nous assurer que la composante IA du système d'enseignement respectait entièrement les règles de la logique mathématique. Le système résultant permet un dialogue intensif et est logiquement à toutes épreuves, puisqu'il est impossible d'étendre la base de connaissances dans des directions contradictoires. D'autres vérifications logiques imposées au système font en sorte que les utilisateurs ne peuvent ajouter à la base de connaissances des données logiquement redondantes; il s'agit là d'une solution qui améliore grandement l'exécution du programme. Des projets existent déjà pour l'implantation en milieu industriel du système d'enseignement assisté par ordinateur auquel nous travaillons. En fait, il sera implanté en Colombie Britannique dans l'usine Prince George Pulp and Paper Mills dans le cadre d'un projet de modernisation de l'usine de 22 millions de dollars.

THE ROLE OF LOGIC IN COMPUTER ASSISTED INSTRUCTION

Jeffrey Skosnik, Ph.D.

British Columbia Institute of Technology
3700 Willingdon Ave.
Burnaby, British Columbia
Canada V5G 3H2

> Classical teaching was chiefly interested in the man himself, not in equipping technicians for specialized jobs; and it is this, perhaps, that most sharply distinguishes it from the education of our own time, which makes it its first aim to produce specialists required by a civilization that to a quite fantastic extent has been invaded by technology and split up into fragments. . . . . There is a terrible tyranny at the heart of technology. Any particular technique tends by its own inner logic to develop exclusively along its own lines, in and for itself, and thus it ends by enslaving the man whom it should serve.
>
> H.I. Marrou, 1956

1. HISTORICAL INTRODUCTION

It has become a commonplace to say that we are in the midst of a second Industrial Revolution. In the first Industrial Revolution, human *physical* labor was replaced by the labour of machines. Today, in the second Industrial Revolution, it is human *mental* labor which is being replaced by the labor of machines. These machines are computers using Artificial Intelligence and Expert System Technology to solve problems normally thought to require human intelligence. More than anything else, it is the use of logic which enables computers to mimic human intelligence. Nevertheless, logic as a subject is sadly neglected by our school system outside the universities. Before dealing specifically with the role of logic in computer assisted instruction, it is appropriate to review the role which logic has historically played in Western culture. Such a review will assist us in forming a clear conception of the role which logic should now play in education.

Deductive reasoning can be either numerical or non-numerical. Story problems from high school algebra are examples of numerical, deductive reasoning. The traditional syllogism is an example of non-numerical, deductive reasoning. The greater part of human deductive thought is probably non-numerical in character--that is, it is logical rather than mathematical.

Though Aristotle is credited with the invention of logic, there is truth in John Locke's remark, "God did not create man and leave it to Aristotle to teach him how to think". What Aristotle did, of course, was to organize logic as a discipline which could be taught as a separate subject. And from Aristotle's time to the Renaissance, logic was an extremely important subject in Western education. The leading intellects of the Renaissance, however, dismissed logic as useless, Scholastic hairsplitting. The Renaissance was a time when people were intoxicated by the discovery of new truths about the world. In his *Dialogues concerning Two New Sciences* (pp. 135-6), Galileo captured the spirit of the times when he said, "Logic, it appears to me, teaches us how to test the conclusiveness of any argument or demonstration already discovered and completed; but I do not believe that it teaches us to discover correct arguments and demonstrations."

The most creative, scientific efforts of the Renaissance were directed toward the development of mathematical techniques which allowed us to grasp the world in a quantitative way. The culmination of this approach was, of course, Newton's theory of gravitation. The quantitative successes of natural science attracted the best minds to mathematics and physics, and eclipsed the further development of logic until the mid-nineteenth century. At that time, deep problems within the foundations of mathematics itself led a small number of gifted thinkers to seek logical solutions to these problems. In seeking solutions to these problems, these thinkers greatly extended the scope of logic, and created a framework which could be used to present any discipline in a clear, consistent manner.

Surveying the history of applied mathematics, one sees that there are periods of intense, creative work in which new ideas enter the currents of thought. Inevitably, however, such periods are followed by more sober times when these ideas must be reconciled and integrated into a consistent body of theory. With their emphasis upon discovery, Renaissance thinkers did not adequately value logic for its ability (in Galileo's words) "to test the conclusiveness of any argument or demonstration already discovered". But, when one thinks of the disputes which have divided mankind, one sees a need "to test the conclusiveness" of arguments.

This is not the time to neglect logic. For, just as the hand-held calculator has automated mathematical calculation, the computer is automating logical calculation. There is an emerging Expert Systems technology which allows the computer to use logic to solve deductive problems in much the way that calculators can be used to solve mathematical problems. When calculators were introduced into schools, there was a debate over whether their use would reduce our appreciation for mathematics. Perhaps the debate has ended; and people today realize that, with proper instruction, the calculator enhances understanding by doing the mechanical work in problem solving while leaving the mind free to focus on mathematical concepts. But what allows the calculator to be used in this way is, of course, that the fundamental concepts of mathematics are taught in school. Unfortunately, the fundamental concepts in logic are not taught in school; and so the users of logic based Expert Systems may be in the position of people trying to use a calculator while not knowing anything about arithmetic apart from what the calculator itself tells them. If we allow this to happen, then Marrou's worst fears about technology (as expressed at the head of this section) will be realized.

2. THE NEED FOR CONSISTENCY AND THE EXPLANATORY VALUE OF LOGIC

The discovery of irrational numbers revealed to the Greeks that their most basic beliefs about numbers and measurement were inconsistent. It seem probable that the Greeks developed logic and the axiomatization of geometry at least partly in the hope of resolving the contradictions in their mathematical outlook. More recently, logic and set theory have been used to provide a clear, consistent foundation for elementary analysis. By definition, an inconsistent theory

cannot possibly be true; therefore, any method which exposes, and helps us to remove, inconsistencies is useful. Therefore, even if (as Galileo alleged) logic does not create new information, it has great value for its ability to transform old information and reveal the connections between our various beliefs. We understand, not by memorizing isolated facts, but integrating new information into an acceptable structure which, as a whole, explains some aspect of reality. Logic is an indispensable tool for determining how new information is to be fitted into the structure of our existing beliefs. Until we have considered new information in light of our other beliefs, we are hesitant to accept that information and do not feel that we understand it fully.

Sometimes, in explaining concepts we use so little logic, that our explanations can at most convey an illusion of understanding. It is a commonplace in elementary school books, for example, to "explain" negative numbers in terms of debts, as opposed to assets. Though such an explanation might be briefly satisfying, a thoughtful student is bound to wonder eventually about the relationship of this explanation to the rules of calculation: a negative number times a negative number equals a positive number. But why should a debt multiplied another debt suddenly turn into an asset? A logical explanation of the nature of negative numbers would also explain the rules of calculation for negative numbers, and remove the sense of mystery about these rules. Once there is an understanding of the rules for calculation, students should use a calculator to do arithmetic; for it has become a mechanical task. Similarly, let us provide students with logical training, and then give them computers for the tedious, mechanical verification of argument. In so doing, we shall free their minds to explore a subject creatively and freely, while providing them with the means of later verifying their conjectures. This is a use of technology which liberates man rather than enslaves him.

3. COMPUTER ASSISTED INSTRUCTION (CAI)

Having defended the use of logic in education generally, let us turn now to its use in computer assisted instruction. People not specializing in CAI tend to know more about the early failures of CAI than about some of its current successes. Admittedly, the early advocates of CAI grossly underestimated the difficulties involved in producing *good* CAI software. Here too history provides us with some interesting lessons: it took a very long time, many centuries, for man to learn how to organize and present information in books. The methods are institutionalized now and well understood. Books are virtually the only model we have for the representation of complex bodies of information. It is not surprising, therefore, that books became the first model for CAI software. But it was a model which reduced the computer to a mere "electronic page turner". Only now are we beginning to understand how the computer can be used in a way which is at once novel and helpful for instructional purposes. Amongst the new approaches to CAI is the one I have adopted. It is an approach which involves using a system of hypertext controlled by logic. Hypertext is simply a data base of text and graphical images which can be non-sequentially accessed.

To understand the role of logic in controlling hypertext, it is useful to consider another data base--namely, the one in our own heads. Human memory is a very interesting phenomenon; and I was much amused once to hear an engineer from Texas Instruments complaining that God had done a very poor job of designing it. Whenever this engineer's wife sent him to the store to buy a dozen items, he would forget half of them; whereas, Texas Instrument computers recall everything stored in them. Why do people forget? There is a very famous case reported by the psychologist Aleksandr Luria of a man who allegedly never forgot anything. This man's life turned out to be a disaster. He worked as a reporter, an occupation for which a perfect memory would appear helpful. Nevertheless, he was a failure since he was unable to select and focus his attention. Virtually everything reminded him of things which directed his attention away from the matter at hand. In this man's life, we see a reason for forgetfulness.

Pure hypertext shares many of the difficulties of Dr Luria's patient. That is, there are no internal controls which dispense with information, and one can move through the data base non-sequentially, randomly. Here I may quote from one vendor's (*Cogent Software Ltd*) hyperBase User Manual (p. 4):

> Hypertext is a very powerful document structuring technique which can increase greatly the efficiency with which a reader can obtain information from a computer-based document, however, it does introduce some of its own problems. Foremost amongst these is the syndrome often referred to as 'lost in Hyperspace'. This results when a reader of the document has been following a long chain of Hypertext links and suddenly finds that he has lost his chain of thought or position in the document. The reader has been tempted to follow the connecting links to the detriment of the main purpose for using the document and has become mired in a collection of details are related issues.

One could, of course, organize a hypertext document into a electronic book--with a table of contents and an index for a sequence of chapters, pages, and paragraphs. This would give the document an intelligible structure, of course; but it would defeat the purpose of using hypertext. What is required instead is a logic component which maintains control over user access of the hypertext. Remember what logic does best is to structure information, present it, explain it, and verify it. (*Cogent's* version of hypertext can be programmed using Prolog, which is a language offering some of the functionality of logic.)

4. APPLICATIONS OF THIS TECHNOLOGY

Foremost amongst the applications of this technology is, of course, its use in Expert Systems. Let us consider what an Expert System is. It is a piece of computer software which somehow captures the knowledge of a human expert, and which is capable of dispensing that knowledge to others independently of the expert from whom it was derived. But no one wants to act on advice which one does not understand. One wants to be able to ask why that advice is given. And there is nothing more satisfying than an explanation which shows how a particular recommendation follows logically from accepted assumptions. Logic can be used to persuade the users of an Expert System that something is true, given the known facts. This makes users accept the advice given by the Expert System.

The kind of Expert System we are dealing with here is not one which advances the frontiers of human knowledge. It is rather a reservoir of existing knowledge, from which we can all draw according to our individual needs. The value of such a system to our educational institutions should be immediately obvious. Logic can remain in the background while students probe a hypertext data base. In the background, logic can perform the highly useful function of presenting explanations as required and controlling the flow of information--in a highly personalized manner--to the individual user. To explain precisely how this happens is beyond the scope of this paper. As noted in the abstract to this paper, however, we have developed a logic based system of hypertext with a *logically* incorruptible knowledge base and a number of other logical safeguards. Over the course of the next year, this system will used at *Prince George Pulp and Paper Mills* as part of an information management system for distributed control. I must praise this company for its farsightedness in deciding to explore AI technology. Similar words of praise apply to *SunRype Products*, in Kelowna, B.C., where we shall also be installing one of our systems.

5. CONCLUSION

Logic can be used very successfully in computer-assisted learning packages, provided the authors of such packages have adequate training in the subject. Logic is becoming an extremely important discipline in itself, for which more training is required than we as a society are presently offering. In 1963 John McCarthy, the author of the programming language Lisp, remarked that, "It is reasonable to expect that

the relationship between computation and mathematical logic will be as fruitful in the next century as that between physics and analysis in the last". One hopes that our educational institutions will adjust to our changing needs, and provide students from a young age with the training in logic necessary for them to share in the future development of computer science.[1]

---

[1]The author gratefully acknowledges the advice and support for his project work which he has received from the following individuals and organizations: (1) Sharlene Arnesen and Greg Tolliday of the CEIC Innovations Program; (2) Gerry Brumpton, SunRype Products Ltd; (3) Dennis Duffey, IBEW/ECA Training Co-ordinator; (4) M. Floeck, President, Pacific Artificial Intelligence Systems Corp.; (5) Brodie Harrington, formerly of the CEIC Innovations Program; (6) Dave Hume, formerly of BCIT; (7) Roy Murray, former president of BCIT, (8) Dave Yarmish, Prince George Pulp and Paper Mills; and (9) British Columbia's Ministry of Advanced Education and Job Training. Finally, the core members of our research team are: N. Hume, P. Kay, M. Litster, J. Skosnik, T. Stokes, and D. Svetic.

REFERENCES

Towards a National Policy for a Computer-Assisted Learning Industry (National Research Council Associate Committee on Instructional Technology, May 1984).

HyperBase User Manual (Version 1.0), Cogent Software Ltd.

Galilei, Galileo, *Dialogues Concerning Two New Sciences* (New York: 1954).

Kleene, S., *Mathematical Logic* (New York: 1967).

Marrou, H., *A History of Education in Antiquity* (New York: 1956). (The quotation at the head of this paper is taken from this book, see pp. 302 and 306.)

Siekmann, J. and Wrightson, G., eds, *Automation of Reasoning* (New York: 1983); 2 volumes. (The quotation from Mccarthy is taken from the backcover of this book.)

TRAINING TECHNOLOGY'S MAGIC MIRROR - THE COMPUTER IN COMPUTER ASSISTED LEARNING

G. Palen
Training Technology Unit
Transport Canada Training Institute
Cornwall, ON

Transport Canada has recently begun to explore the application of Computer Assisted Learning technology to our technical training environment. We are accustomed to teaching operational personnel how to teach. This is the Technology Unit's main mission. Now, we are beginning to learn how to 'teach' computers to teach. This has presented us with a very different problem.

As we 'teach' the computer to deliver instruction, it has forced us to better define our instructional strategies; as we teach the computer to assist our students in learning, it has forced us to deepen our understanding of student learning processes and as we 'teach' the computer to make more intelligent responses to student input, we are forced to learn more about intelligent feedback strategies. The computer requires us to make explicit the intelligence we want it to use in the teaching encounters. While the predicted gains in effectiveness and efficiency resulting from implementing CAL in the Department of Transport are the goals which drive the introduction of this medium, this Magic Mirror effect may have even greater impact on our overall effectiveness and efficiency. Our deeper understanding of instructional strategies, learning strategies, and intelligent feedback are stimulating us to modify our existing classroom-based courses to take these new ideas into account. So, although CAL is still in the early implementation phase, the benefits of CAL are already being seen in our traditional instruction.

Some examples will be discussed which show the effect of such implementation.

MIROIR MAGIQUE DE LA TECHNOLOGIE DE LA FORMATION - L'ORDINATEUR DANS L'ENSEIGNEMENT ASSISTÉ PAR ORDINATEUR

Transport Canada vient de commencer à étudier l'application de la technologie de l'enseignement assisté par ordinateur à ses cours de formation technique. Nous avons l'habitude de montrer à nos enseignants comment enseigner. C'est là la principale mission de la Section de la technologie de la formation. Nous commencons actuellement à apprendre comment montrer aux ordinateurs à enseigner. Les problèmes auxquels nous avons dû faire face étaient très différents.

Le fait de montrer à l'ordinateur à transmettre ses connaissances nous a forcés à mieux définir nos stratégies d'enseignement; en présentant à l'ordinateur la façon d'aider nos étudiants à apprendre, nous avons dû approfondir notre connaissance des méthodes d'apprentissage de l'étudiant. Enfin, en montrant à l'ordinateur à répondre plus intelligemment aux questions des étudiants, nous avons été obligés d'en apprendre davantage sur les stratégies de rétroaction intelligentes. L'ordinateur nous oblige à expliciter l'intelligence que nous voulons qu'il utilise dans des séances de formation. Outre les gains d'efficacité et de rentabilité qu'on prévoit réaliser grâce à l'application de l'EAO au Ministère des Transports et qui nous ont poussés à adopter ce medium, le Miroir Magique peut avoir une plus grande incidence encore sur notre efficacité et sur notre rentabilité globales. L'approfondissement de nos connaissances en matière de stratégie d'enseignement et d'apprentissage, et de rétroaction intelligente, nous incite à modifier nos cours de base existants et à tenir compte de ces nouvelles idées. Même si l'EAO vient tout juste d'être mis en application, les avantages qu'il offre sont déjà perceptibles au niveau de l'enseignement classique.

Nous discuterons de certains exemples qui illustrent les répercussions de l'utilisation de cette application.

TRAINING TECHNOLOGY'S MAGIC MIRROR - THE COMPUTER IN COMPUTER ASSISTED LEARNING

G. Palen
Training Technology Unit
Transport Canada Training Institute
Cornwall, ON

(MANUSCRIPT NOT AVAILABLE AT TIME OF PRINTING)

(MANUSCRIT NON DISPONIBLE)

AN INSTRUCTION/PRODUCTION MODEL FOR TRAINING COURSEWARE

J. Vazquez-Abad, J. LeCavalier
Les Technologies EDINOV Inc.
Montréal, PQ

The main objective of this presentation is to discuss some problems related to the industrial production of training courseware. The particular case of Montréal-based EDINOV Inc. will be discussed. EDINOV is a Canadian firm created around a CBT project on waste- and drinking-water treatment plants which attracted a multidisciplinary team, one with partcular strengths in management as well as in technical aspects (instructional design, computer programming and graphic design). The project entailed the production of courseware for the 45-course curriculum for plant operators; it required, as the first activity, the development of an adequate instructional design and the implementation of a production model for that design. The more elaborate interplay between task- and target-population analyses, choice of technology (hardware, software, instructional approach), instructional design (which includes a learner-control strategy enhanced with advance organizers, sophisticated drill and practice and entailment mesh-based dictionaries designed to both help students with lesson content and allow learning by browsing), production model and managerial strategies will be discussed, as well as the R&D stages, including the prototyping of both instructional and production models. Examples of training courseware thus produced will be shown at the presentation.

MODÈLE D'INSTRUCTION ET DE PRODUCTION D'UN DIDACTICIEL DE FORMATION

Le présent exposé a pour principal objet de traiter certains des problèmes reliés à la production industrielle des didacticiels de formation, et aborde le cas particulier de la société EDINOV Inc., basée à Montréal. Il s'agit d'une société canadienne mise sur pied dans le cadre d'un projet d'EAO axé sur des stations de traitement de l'eau potable et des eaux usées, qui a attiré une équipe multidisciplinaire regroupant surtout des personnes très compétentes en gestion et en questions d'ordre technique (stratégie de développement, programmation et graphisme). Le projet portait sur la production d'un didacticiel destiné au programme d'études de 45 cours des opérateurs de station; il fallait en premier lieu, proposer une stratégie de développement appropriée et mettre en oeuvre un modèle de production correspondant à cette stratégie. Le document traite des rapports complexes entre la tâche et les études des groupes cibles, des choix technologiques (matériel, logiciel et approche pedagogique), de la stratégie de développement (comprenant une stratégie de direction des étudiants enrichie de textes préliminaires, d'exercices évolués, de dictionnaires par recoupements permettant aux élèves de mieux saisir le contenu du cours et de mieux apprendre par la lecture), d'un modèle de production et des stratégies de direction, ainsi que des étapes de recherche et de développement, y compris la création de modèles pédagogiques, et de production. Des exemples de didacticiels de formation créés selon cette approche seront présentés dans le cadre de cet exposé.

AN INSTRUCTION/PRODUCTION MODEL FOR TRAINING COURSEWARE

Jesús Vázquez-Abad[1] and Jacques LeCavalier[2,3]

Les Technologies EDINOV Inc., 417 St Nicolas,
bureau 200, Montréal (Québec) Canada H2Y 2P4.

The main objective of this paper is to discuss some aspects of the industrial production of courseware; thus, it will adopt a producer's point of view --one which strives for profitability as the way to ensure the survival of the firm. The particular case of Montreal-based EDINOV INC. will be used to illustrate the discussion.

Educational computing, both in training and school settings, has come a long way since the pioneering attempts of the 60's. In its many forms, it has permeated the educational curricula of most developed countries from preschool to postgraduate levels. Even the conservative area of training is slowly but surely succumbing to the enchantment: according to Training's annual survey, 1987 marked the "coming of age" of CBT in the United States, where in a market dominated by traditional lecture and linear-video instruction, the computer has enticed more than half of the big companies.

However, educational computing is far from being securely implanted. For example, there is never "enough" and/or "adequate" software in many educators' and trainers' opinions, which makes it "impossible" for them to use CBL or CBT. It has also become a commonplace, albeit justified, to complain about the difficulty of quickly and effectively producing quality software for specific instructional purposes. It is thus reasonable to propose that an effort be made to find efficient ways to produce it, concurrent to the search for what makes quality software and for what the impacts of its use are.

It is important to distinguish between three interconnected activities: research, development, and (industrial) production of courseware. Naturally, the first of these gets the most attention from academics. The second one is the realm of many government centres, of big R&D companies, of teachers' consortia and even of home-based producers and it has been given publicity accordingly. But for companies interested in running a profitable operation in educational computing, the focus must be on industrial production; and on this we suffer from a relative dearth of information.

EDINOV is a Canadian firm created around a CBT project on waste- and drinking-water treatment plants which attracted a multidisciplinary team, one with particular strengths both in management (including marketing and international transfer of technology) and technical aspects (instructional design, computer programming and graphic design). This project entailed the production of courseware for the 2-year, 45-course curriculum of government-run vocational education of plant operators; it required, as the first activity, the development of an adequate instructional design and the implementation of a production model for that design.

Briefly, the scope of the project suggested the use of a single design for the mass-production of scripted courseware; the nature of the curriculum seemed best served by a learner-control strategy enhanced with advance organizers and sophisticated drill & practice; the characteristics of the target population led to the development of a highly conversational tutorial style and an iconic, user-friendly interaction code; and the particular flair (or whim) of the first author (then Head Instructional Designer) resulted in the implementation of "paskian" entailment mesh-based dictionaries to both help students with lesson content and allow learning by browsing.

The problem of choosing or developing and establishing a production model for this project is more complex than it seems at first. Among other things, the choice of a production model is closely related to, if not determinant of, the choice of (project) management model one will follow. The literature on courseware development and production and their management, although

---

1 Directeur technique.

2 Instructional Designer.

3 The authors wish to acknowledge the contribution of all the members of EDINOV, as well as of M. Gabriel Lefebvre (*Directeur de la production* at the time it started), in the development and implementation of the production models.

existent, is rather scanty. In fact, its underlying ideas seem closer to the ones stemming from the area of software engineering than from management and administration. These latter claim the existence of concepts, principles nad rules to better manage an organization. In particular, Project Management borrows some of this "good management" knowledge: ideas such as management by objectives, planning, programming, scheduling, budgeting, control, communication and motivation are supposed to apply to (and help in) any project.

However, experience in technical areas such as software engineering shows that a fairly sound knowledge of the particular project domain is needed to undertake its management successfully. A typical project in software engineering entails the definition, development and maintenance of software, such as the operating system of a new computer. Regardless of the precise engineering model followed (Classic Life Cycle, Prototyping, 4GL-Prototyping, etc.), this project may involve a large number of resources (for a probably large period of time) which are nevertheless quite homogeneous: material resources are mainly hardware, human ones may range in competencies but always within the same domain.

This is hardly the situation with courseware production, at least in EDINOV's situation. There was a need to call in knowledge and skills from different domains --with an added complexity due to different professional attitudes, objectives and personalities: programmers and analysts typically at ease with schedules, weekly objectives and controls but often incapable of meeting their own deadlines; graphic artists, communication and media specialists typically uncomfortable with detailed plans and schedules, but often capable of meeting imposed deadlines; instructional designers turned into bargainers, negotiators and even conflict-mediators (involving skills which were certainly not formally taught at school).

Our approach to courseware production for this project took these facts into consideration. As a starting point, a conceptual framework was developed: since training software is, first and foremost, instructional in nature, an instructional model was developed. The constraints taken into consideration for this were (1) the *delivery* context (needs analysis, target population, curriculum and contents, instructional task analysis, target delivery system, special client expectations), and (2) the *production* context (an industrial production one, thus striving for homogeneity and uniformity of both the process and the products). Other constraints included the orientation of the company (systematic design of training, interactive instructional strategies, electronic --mainly informatic-- media), and the particular bent of the main *responsables* for the project.

Thus the model shown in Figure 1 was developed, within the structural context for the general curriculum shown in Figure 2. This model is complemented by a simple yet powerful iconic code for user-courseware interaction, an option offering information on the learning environment, a structure (dictionary) offering conceptual help, a modular division allowing an easy circulation between sections, and a uniform visual design with balanced and well thought-out graphic content reinforcing accurate and concise textual information written in simple and direct language. This model, then, establishes the environment or "shell": the individual courseware production activity can be re-stated as "filling" the shell with the specific content of the course. In turn, this re-statement determines the activities required (and the management thereof) to fill one shell. For example, good Instructional Design principles dictate that, after objectives for the course are determined, test items are to be developed. These items, that will be used in the drill and practice and evaluation sections, should be checked for congruency against the objectives, for accuracy against content knowledge, and for linguistic correctness. Any graphical components should be developed and checked, and integrated with the textual components into the appropriate informatic structure. This obviously calls for the intervention of different human resources, requiring some management to ensure its flow. On the other hand, the *development* of the overall (e.g. menus) and the specific (e.g. drill and practice algorithm) structures, of specific production tools (e.g. overlay techniques, actual icons, storyboards, etc.) and managerial instruments (e.g. production charts, weekly time-use sheets, control and testing procedures) required a different level of elaboration.

The company's services (Consultation, Development and Production) are provided through the three divisions (Instructional Design, Informatics and Media) under the technical direction. The first production model tested established teams supported by division heads. The division heads worked first as a "model team" on a prototype to tune up the instructional and production models, to develop managerial strategies and tools, as well as to improve technical skills in their respective areas of expertise. In a typical prototyping approach, product specifications are loosely determined before the actual development starts, and the prototyping cycle(s) help improve specifications through work with real constraints. As we have stated before, a courseware project differs from a software engineering one, among other things, in the variety of professional skills and areas involved. Thus, prototyping for us started with a thorough study of the instructional problem at hand, and a set of instructional decisions that became parameters to be respected in the search for a "solution." In a sense, this prototyping could be envisaged as an iterative process to find a solution to a

production problem within a (bounded) area to which one has arrived (mainly) via instructional considerations. Once in that area, iterations are caused by the interplay of other technical and practical variables, and attention is paid to restrict the effects of these prototyping iterations mainly (ideally, almost exclusively) to these variables. After a six-month period, the instructional model was refined and pruned, the production model was tested, and most production and communication tools were developed and tuned-up; and the prototype was discarded as is the fate of any such thing.

After a training period, the first production phase started with each of the teams undertaking a course assigned to them by the (then) Production Director; any special technical support needed was the responsibility of the respective division head (see Figure 3). The first courses produced under this model took about nine person-months (3 people times 3 months) each to produce.

Although the products (three complete courses) of this phase proved (highly) instructionally efficient in (subject-matter expert and target population based) formative evaluation, the process evaluation of this phase showed a lack of efficiency in the production of courseware under this scheme, as well as redundancy in developmental tasks and high levels of dissatisfaction and stress. These problems were traced back to the artificial separation of similar tasks and information imposed by the model. We therefore moved towards a division-centered, batch/mode production for the second phase (see Figure 4). Production of each course is handled as a project, whose manager (by now, the Technical Director) divides it into tasks and then requests and coordinates specialised services from the divisions in order to accomplish them. In turn, division heads manage internally the assigned tasks and assure a high degree of intra- and inter-division communication flow. In addition to reducing production times, this model further allows for a greater flexibility *vis-à-vis* the ever important development of the company's human resources and services offered.

While Edinov's experience with the industrial production of courseware remains limited, it is becoming increasingly clear to us that our efforts in this area could contribute to the development of a "generic" instruction/production model. Such a model, which to our knowledge cannot be found in the current literature, would have its basis in organizational and educational cybernetics. It would thus likely be comprised of the following key elements:

- project goals (related to quality, delivery timelines, cost effectiveness, organizational and personnel development, etc.);
- inputs or resources (related to the instructional, production, and delivery contexts of the project, and including such sub-elements as initial content structure, development software and staff competencies);
- production system variables (such as task allocation, work group structure, project management tools, etc.);
- communication processes (within the organizational context of the company and project);
- control processes (related to quality, uniformity, time and financial constraints);
- and finally, outputs or results.

Against the background of these basic elements, the model could be "set in motion" by reproducing, with different "values" of the production system variables, a number of possible production scenarios. These would include the two approaches briefly described earlier (the course team method and the batch/mode production approach). Subsequent analysis of model performance with the aid of conventional cybernetic approaches (feedback loops, requisite variety, etc.), according to desired outputs or results, would complete the cycle.

The beginning attempts at defining and applying this generic model will be presented during the poster session.

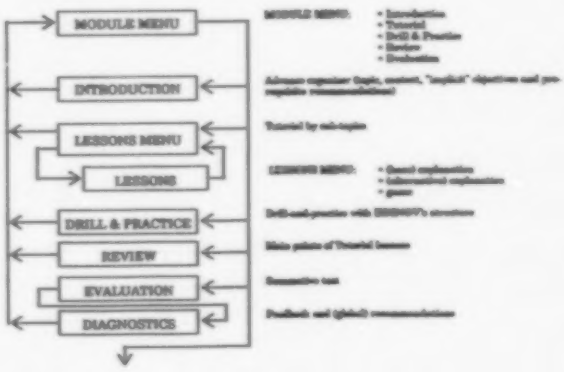


Figure 1
Module Structure

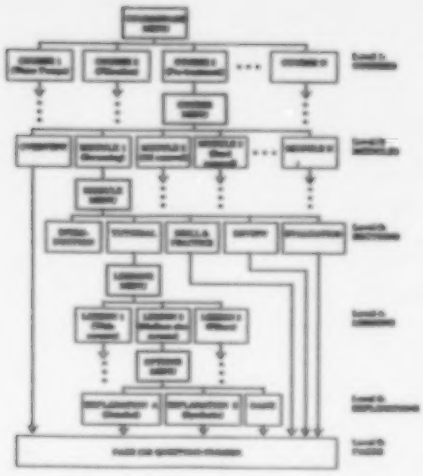


Figure 2
Course Structure and Levels

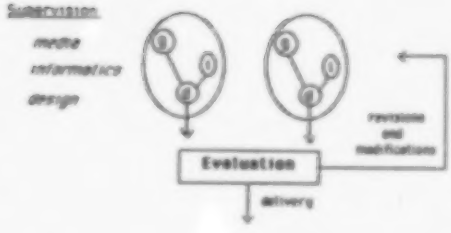


Figure 3
Team Production Model

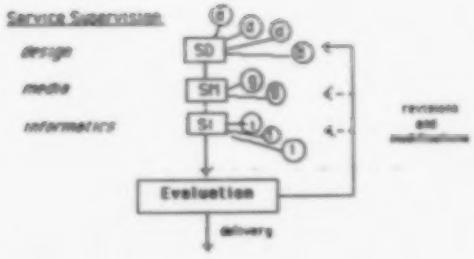


Figure 4
Service (or Batch) Production Model

Bibliography

Betlach, JA (1986) Framework for Computer Assisted Instruction Courseware: A Case Study, in Journal of Educational Technology Systems, 15(1), 35-52

Glen, JA and Walker, LJ (1985) Software Development of Interactive Microcomputer Programs used in Teaching Mathematics: A Case History, in Computers and Education, 9(3), 183-193

Howes, HF and Williams, ED (1986) A Methodology for Developing Computer-Based Teaching Programs, in Computers and Education, 10(3), 347-352

Kontos, G (1984) Instructional Computing: In Search of Better Methods for the Production of CAI Lessons, in Journal of Educational Technology Systems, 13(1), 3-14

Moonen, J (1986) Toward an Industrial Approach to Educational Software Development, in Bork, A and Weinstock, H (eds.) Designing Computer-Based Learning Materials. NATO ASI Series, Vol F23. Heidelberg, Springer-Verlag Berlin, pp 119-151.

Moonen, J and Schoenmaker, J (1986) Production Techniques for Computer-Based Learning Material. Paper presented at the Annual Meeting of the American Educational Research Association. San Francisco, CA, April 16-20. (ED 276 406)

Murphy, PA (1985) A Model for Improving Assessment Techniques Used in Courseware Production: Proposal for Change in Courseware Production Team Structure. Paper presented at the Annual Meeting of the National Council of Measurement in Education. Chicago, IL, April. (ED 255 562)

Raven, BK (1986) Managing Computer-Based Training Projects, in Interactive Learning International, 3(4), 7-14

Training, The Magazine of Human Resources Development. (Since 1963). Published monthly by Lakewood Publications Inc., subsidiary of the Maclean Hunter Publishing Company. Octobre 1987 (Vol. 24, No. 10).

Vázquez-Abad, J. and Marc LaFleur (in press) Design of a Performance-Responsive Drill and Practice Algorithm, in Computers and Education.

Watson, DM (1983) A Model for the Production of CAL Material, in Computers and Education, 7(3), 167-176

Wilson, BN and McCrum, E (1984) Use of Modular Design in the Production of Portable CAL Software: A Case Study, in Computers and Education, 8(2), 229-237

LA FORMATION DES AGENTS D'IMMIGRATION À L'HEURE DU VIDÉO INTERACTIF

L. Desmarais
Direction du développement et de la planification des ressources humaines
Emploi et Immigration Canada
Hull, PQ

Le Ministère de l'Emploi et de l'Immigration a mis sur pied un projet pilote de formation assistée par ordinateur. Ce projet comportait entre autres, l'élaboration et la production de trois didacticiels dont un requérait l'utilisation du vidéodisque géré par ordinateur. Ces didacticiels visaient la formation des agents d'immigration.

Dans cette communication, nous passerons en revue les phases de l'élaboration du didacticiel qui comprennent la production d'un vidéodisque de niveau 3, de l'essai par des représentants du groupe-cible et des considérations techniques qui nous ont permis la réalisation de ce produit avec des moyens financiers limités. Ce didacticiel visait l'utilisation du billet d'avion comme outil d'aide à l'examen d'immigration.

Les participants pourront également visionner le produit et interagir avec l'ordinateur.

TRAINING IMMIGRATION OFFICERS IN THE AGE OF INTERACTIVE VIDEO

The Department of Employment and Immigration has established a pilot project in computer-assisted training. Among other things, this project has involved the development and production of three courseware packages, one of which requires the use of a computer-controlled vidodisc. The purpose of these courseware packages is the training of immigration officers.

In this paper, I will review the phases in the development of the courseware, including the production of a level 3 videodisc, testing by representatives of the target group and technical considerations that have allowed us to implement this project with limited financial resources. The objective of this courseware package is the use of the airline ticket as a tool to assist in the immigration inspection process.

Participants can also view the product and interact with the computer.

LA FORMATION DES AGENTS D'IMMIGRATION À L'HEURE DU VIDÉO INTERACTIF

Lise Desmarais
Division de la formation

Ministère de l'emploi et de l'immigration

En 1986, le Ministère de l'emploi et de l'immigration a mis sur pied un projet pilote de formation assistée par ordinateur. Ce projet comportait entre autres, l'élaboration et la production de trois didacticiels à l'intention des agents d'immigration

APPROCHE DIDACTIQUE

C'est une approche didactique par opposition à une approche technologique qui a présidé au choix du médium d'apprentissage. Ainsi, dans cette approche, la technologie appuie l'apprentissage, les éléments à apprendre sont identifiés à la suite de l'analyse des besoins des apprenants et des moyens d'évaluation permettent de vérifier les apprentissages et les résultats de l'expérience.

Dans une approche technologique, on procède à l'inverse. On choisit une technologie et on en fait l'utilisation. Le but d'une telle approche est l'exploitation de la technologie.

DETAILS DU PROJET

Choix de la technologie

L'un des trois didacticiels visait à permettre l'utilisation du billet d'avion comme outil d'aide à l'examen d'immigration. Ainsi, au terme de l'apprentissage, l'agent examinateur doit être en mesure de distinguer des incohérences entre les données du billet et la déclaration d'un voyageur.

Pour ce type d'apprentissage, l'ordinateur offrait des possibilités intéressantes en permettant l'utilisation d'une banque de billets, le jugement immédiat des réponses, le choix de cheminements particuliers selon les besoins et les connaissances des apprenants. De plus, l'utilisation du son et de l'image apparaissait essentielle. Il fallait que l'apprenant soit en mesure de voir des billets d'avion et d'entendre les déclarations des voyageurs. Le vidéo interactif était donc tout à fait approprié.

Elaboration

L'élaboration de ce didacticiel s'est faite en consultation avec un expert en billetterie de l'Association internationale du traffic aérien (IATA) afin de vérifier les éléments de contenu.

Nous avons opté pour une approche analytique dans le traitement du contenu Ainsi, les éléments d'apprentissage sont définis et proposés aux apprenants avant d'être amalgamés dans une activité de simulation. Par ailleurs, les apprenants ont le choix des cheminements et peuvent opter d'abord pour la simulation avant de passer à l'étude des divers éléments.

La variété méthodologique a été une préoccupation. Nous voulions offrir aux apprenants diverses modalités d'apprentissage de façon à soutenir leur intérêt et faciliter l'apprentissage.

Pré-production

Une fois le prototype terminé, le contenu a été validé une fois de plus par l'expert en billetterie, la conception du programme et la méthodologie par des représentants du système-client et l'utilisation de la technologie par les responsables de la programmation. Une phase de révision a permis l'intégration des commentaires et recommandations des divers intervenants.

Production

Une compagnie de production (SOFI) a assumé la production des enregistrements en studio ainsi que les diverses étapes d'édition précédant le matriçage du vidéodisque.

Afin de rentabiliser l'utilisation du vidéodisque, nous avons opté pour le système de fenêtres permettant de doubler l'utilisation de la même piste image. Les deux pistes sonores sont utilisées pour le français et l'anglais.

Puisqu'il s'agit d'un vidéodisque de niveau 3, la gestion des images et du son se fait à l'aide de l'ordinateur et le vidéodisque offre un banque de données gérées par le programme informatique.

Evaluation

Un processus d'évaluation du didacticiel a été mis en place afin d'évaluer les apprentissages. Un pré-test et un post-test identiques ont été administrés afin de déterminer si l'utilisation de ce didacticiel assurait l'apprentissage des éléments de contenu.

Il ressort de cette expérience que ce moyen didactique facilite l'apprentissage de façon significative. De plus, les attitudes des participants sont très favorables à l'utilisation de cette technologie.

Mise en oeuvre

A la suite de l'évaluation, le didacticiel a été revisé et mis à la disposition des agents d'immigration.

A ce jour, les agents d'immigration de l'aéroport de Vancouver ont utilisé ce didacticiel à des fins de formation. Depuis janvier, les agents d'immigration à l'aéroport d'Ottawa en bénéficient.

Le Ministère a acquis cinq ensembles itinérants (ordinateurs, moniteurs, lecteurs de vidéodisque) pour utilisation à des fins de formation selon la demande des employés.

PROSPECTIVE

A la suite d'expériences dans le domaine, le Ministère de l'emploi et de l'immigration, par l'intermédiaire de la Division de la formation a décidé de considérer la formation assistée par ordinateur comme une option méthodologique à explorer lorsque des demandes de formation des employés lui sont addressées.

Une analyse préliminaire du contexte situationnel du système-client ainsi qu'une étude coût-bénéfice seront les moyens qui permettront de déterminer si cette option méthodologique constitue une option valable pour le type de formation souhaitée.

AUTHOR INDEX/INDEX DES AUTEURS

ABBOTT, S. 481
ALLEN, M. 124
AL-YAMANI, S. 345
ANWAR, A.K. 377
ARULLANI, A. 222
BARBEAU, L. 80
BARKER, P.G. 377
BARONIAN, A. 54
BEAM, P. 243
BEAUCHEMIN, S. 519
BEAULIEU, H. 54
BÉLANGER, J.-C. 54
BELL, D. 377
BELL, W. 412
BENGIO, S. 133
BENSLIMANE, A. 114
BHUIYAN, S. 49
BIERMAN, D.J. 457
BODEY, M.R. 273
BOILY, A. 155
BORNE, I. 402
BOTTINO, R.M. 185
BOULET, M.-M. 80
BOYD, G.M. 200
BOYER, H. 397
BRODERICK, W.R. 433
BRODY, M.J. 119; 212
BROWN, R.H. 75
BRULÉ, G. 44
BUCK, G.H. 360
BUELL, A.M. 129
BURSTON, G.V. 515
CAMERON, J. 422
CAMIRAND, C. 486
CAPPELLO, G. 222
CARRIER, S. 486
CARTWRIGHT, G.F. . . . . 99 ; 337
CAUZINILLE-MARMECHE, E. 58
CHANG, C.S. 465
CHARRON, R. 99
CHRISTENSEN, L.C. 273
CLARK, H.C. 1
CLARK, R.J. 68
CODERRE, W.M. 14
COFFIN, L. 337
COULMAN, R. 49
COULOMBE, A. 479
COULOMBE, C. 479
CRAPPER, C.G. 515
DALKIR, K.L. 70
DE LA PASSARDIÈRE, B. 39; 305
DE PINTO, M. 222
DÉCARY, M. 479
DECORTE, M. 382
DÉRY, D. 310
DERYCKE, A. 285
DESAUTELS, P. 479
DESAUTELS, R. 479
DESMARAIS, L. 534
DORAN, L.A. 427
DUCATEAU, C.F. 114
DUCHASTEL, P. 310
DUFRESNE, A. 315
DUMONT, B. 143
DURELL, B. 341
ELIE, F. 85
ELSHOUT, J.J. 457
EMMOTT, L.C. 390
FAHY, P.J. 364
FOTHERGILL, R. 433
FRASSON, C. 39; 133
FRECHETTE, J.-Y. 505
FRITZ, J.M. 234
FUGÈRE, B.J. 90
FUGLER, D. 238
GABBIANELLI, C. 222
GAGNÉ, M. 148
GAUTHIER, G. 109
GECSEI, J. 133
GIRARDOT, C. 402
GIROIRE-BROUSSE, H. 226
GLADWELL SCHULZE, K. 63
GODFREY, W.D. 129
GRAY, A. 160
GREER, J. 49
GREVE, S.H. 164
GRIFFIN, D.K. 443
GRIFFITH, A.I. 369
HABIB, M. 85
HAEUSLER, E.H. 195
HAMEL, R. 505
HARRI-AUGSTEIN, E.S. 412
HAYHOE, W.D. 268
HAYS, W. 273
HEATH, J. 85
HENRI, F. 290
HICKMAN, C. 160
HOLLINGSWORTH, M. 470
HOPE, P.H. 501
HUNKA, S. 29
HYWORON, Z.L. 448
IMBEAU, G. 109
IMBEAU, J. 310
JACQUES, A. 148
JOAB, M. 58
JONES, M. 104

JORDAN, G.A. 234
KALINOWSKI, M.F. 427
KATZ, L. 475
KAYLER, H. 262
KIBBY, M.R. 320
KNUSSEN, C. 320
KOPEC, D. 212
LAMONTAGNE, C. 397
LAPOINTE, J. 155
LAROSE, S. 207
LAVOIE, M. 148
LE BLANC, C. 238
LE XUAN, A. 138
LEBLANC, M.D. 190
LECAVALIER, J. 529
LEE, Y.P. 465
LEES, B. 217
LESCOP, J.-Y. 290
LÉVESQUE, G. 44
LEVIN, T. 278
LIU, J. 200
MARCOS, B. 155
MARK, M. 49
MAROM, R. 358
MASON, R. 422
MATHIEU, J. 58
MAYES, J.T. 320
McALEESE, R. 7
McCALLA, G. 49
McGARRY, M.A. 119
McLEAN, R.S. 437
MEILLEUR, D. 173
MEURRENS, M.W.F. 168
MITCHELL, P.D. 34; 173
MONAGHAN, R.D. 258
MORISSETTE, J. 382
MORRILL, M. 481
NACHMIAS, R. 278
NESBIT, J.C. 29
NIEM, L. 90
NUTBROWN MASSEY, S. 353
O'KELLY, P.J. 295
PALEN, G. 527
PATTERSON, B. 119
PELCHAT, R. 417
PEPIN, M. 207
PHILLIPS, S.E. 253
PLUME, J. 510
POISSON, D. 285
PRASAD, B. 49
QUIGLEY, M.T. 95
RANSOM, R. 303
RICHARDS, G. 386
ROBERGE, M. 397
ROSS, M.L. 493
SALL, H.N. 453
SANDALS, L.H. 180
SCHMALHOFER, F. 99
SCHWEIGHARDT, M. 49
SHARON, D. 358
SHERIDAN, D.P. 351
SHINGHAL, R. 138
SHINGLES, R. 129
SIEMENS, R. 243
SKOSNIK, J. 523
SLEIGHTHOLM CAIRNS, B.J. 295
SLOBODRIAN, S. 80
SMITH, L.R. 327
STENTON, D.E. 264
ST-PIERRE, F. 407
SWALLOW, J. 205
SZABO, M. 124; 360
TALBOT, R.J. 497
TAYLOR, R. 364
THÉRIEN, L. 44; 155
TRAVERS, J. 20
TREMBLAY, R. 90
VAN DER KOOIJ, W. 462
VAZQUEZ-ABAD, J. 529
VEENMAN, M.V.J. 457
VEILLETTE, M. 155
VICKERS, J.N. 481
VIEVILLE, C. 285
WAN, J.-R. 231
WATTERSON, P.R. 248
WEBER RUSSELL, S. 190
WEBSTER, J. 234
WELCH, G. 258
WINER, L.R. 258
WIPOND, K. 104
WITWICKI, G. 475
WOLTER, H. 364
WU, M.-J. 231
WYATT, K. 510
YOUNG, M. 332
YOUNG, R. 479